电气信息工程丛书

# 西门子 PLC、变频器、触摸屏工程应用及故障诊断

向晓汉 主编





本书配有60个源代码和8个视频文件

下载网址为 www.cmpbook.com



机械工业出版社
CHINA MACHINE PRESS



VISIT…

LANZAROTE Caliente.COM

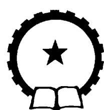
电气信息工程丛书

# 西门子 PLC、变频器、触摸屏工程应用及故障诊断


主　编　向晓汉

副主编　曹英强

主　审　胡俊平


机 械 工 业 出 版 社

本书从基础和实用出发，主要内容包括 S7-300/400 可编程序控制器、变频器和触摸屏。全书分两个部分，第一部分（第 1 ~ 7 章）为入门学习，主要介绍 S7-300/400 的硬件和接线、STEP 7 软件的使用、PLC 的编程语言、编程方法与调试；第二部分（第 8 ~ 13 章）为工程应用，包括 PLC 的通信、PID 技术、HMI 技术、变频器调速、故障诊断和工程应用。

本书内容丰富，重点突出，强调知识的实用性，几乎每章中都配有大量实用的例题，便于读者模仿学习。大部分实例都有详细的软件、硬件配置清单，并配有接线图和程序。本书的资源中有重点内容的程序和操作视频资料。

本书可供学习 S7-300/400 入门和提高级的工程技术人员使用，也可以作为大中专院校的机电类、信息类专业的教材。

**图书在版编目（CIP）数据**

西门子 PLC、变频器、触摸屏工程应用及故障诊断/向晓汉主编．—北京：机械工业出版社，2017.7
（电气信息工程丛书）
ISBN 978-7-111-58358-5

Ⅰ．①西…　Ⅱ．①向…　Ⅲ．①PLC 技术②变频器③触摸屏
Ⅳ．①TM571.61　②TN773　③TP334.1

中国版本图书馆 CIP 数据核字（2017）第 263068 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）
策划编辑：时　静　　责任编辑：时　静
责任校对：张艳霞　　责任印制：孙　炜
保定市中画美凯印刷有限公司印刷
2018 年 1 月第 1 版 · 第 1 次印刷
184mm × 260mm · 37.75 印张 · 930 千字
0001 - 2500 册
标准书号：ISBN 978-7-111-58358-5
定价：129.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
电话服务　　网络服务
服务咨询热线：（010）88361066　　机 工 官 网：www.cmpbook.com
读者购书热线：（010）68326294　　机 工 官 博：weibo.com/cmp1952
（010）88379203　　教育服务网：www.cmpedu.com
**封面无防伪标均为盗版**
金 书 网：www.golden - book.com

# 前　言

可编程序控制器（PLC）、变频器和触摸屏（HMI）被称为“工控三大件”，已经广泛应用于工业控制。目前市面上，关于可编程序控制器应用技术、变频器应用和触摸屏都分别有对应的图书。我们考虑到“工控三大件”通常应用于同一个控制系统，是一个有机的系统，不宜人为分割，所以将可编程序控制器、变频器和触摸屏合并成一本书，这种做法更加切合工程实际。

由于西门子 PLC、变频器和触摸屏具有卓越的性能，因此在工控市场占有非常大的份额，应用十分广泛。虽然 S7 – 300/400 系列 PLC 被大多数技术人员接受，但长期以来，S7 – 300/400 系列 PLC 一直公认是比较难入门的。故本书力求简单、详细，用较多的例子引领读者入门，让读者读完入门部分后，能完成简单的工程。应用部分精选工程的实际案例，供读者模仿学习，以提高读者解决实际问题的能力。为了使读者能更好地掌握相关知识，我们在总结长期的教学经验和工程实践的基础上，联合相关企业人员，共同编写了本书，力争使读者通过“看书”就能学会 S7 – 300/400 系列 PLC、变频器和触摸屏。

PLC 控制系统的故障是难点，本书除了讲解常用的故障诊断的方法，还列举了 30 个常见的例子，便于读者学习、分析。

在编写过程中，本书融入了一些生动的操作实例，提升了读者的学习兴趣。本书与其他相关书籍相比，具有以下特点：

1）用实例引导读者学习。本书的大部分章节用精选的例子讲解。例如，用例子说明现场总线通信的实现的全过程。

2）重点的例子都包含软硬件的配置方案图、接线图和程序，而且为确保程序的正确性，程序已经在 PLC 上运行通过。

3）对于比较复杂的例子，配有录像资料。如工业以太网通信的硬件组态较复杂，就配有组态过程的录像资料，便于读者学习。

4）实用性强，本书的实例易于进行工程移植。

本书由向晓汉任主编，曹英强任副主编。无锡职业技术学院的胡俊平任主审。

全书共分 13 章。第 4、5、8 及 12 章由无锡职业技术学院的向晓汉编写；第 1 章由四川省华蓥市经济和信息化局唐克彬编写；第 2 章由无锡雷华科技有限公司的欧阳思惠和陆彬编写；第 9 章由无锡雪浪环保科技有限公司的王飞飞编写；第 6 章由无锡职业技术学院的林伟编写；第 3 章由无锡雪浪环保科技有限公司的刘摇摇编写；第 7、10 章由桂林电子科技大学的向定汉编写；第 11、13 章由无锡雪浪环保科技有限公司的曹英强编写。参与编写的还有李润海、苏高峰等。由于编者水平有限，书中不妥之处在所难免，敬请读者批评指正，编者将万分感激！

作　者

2017. 1

# 目　　录

前言
第 1 章　可编程序控制器（PLC）基础 …… 1
1.1　概述 …… 1
1.1.1　PLC 的发展历史 …… 1
1.1.2　PLC 的主要特点 …… 2
1.1.3　PLC 的应用范围 …… 3
1.1.4　PLC 的分类与性能指标 …… 3
1.1.5　PLC 与继电器系统的比较 …… 4
1.1.6　PLC 与微机的比较 …… 4
1.1.7　PLC 的发展趋势 …… 5
1.1.8　PLC 在我国 …… 5
1.2　PLC 的结构和工作原理 …… 6
1.2.1　PLC 的硬件组成 …… 6
1.2.2　PLC 的工作模式 …… 10
1.2.3　PLC 的工作原理 …… 10
1.2.4　PLC 的立即输入、输出功能 …… 12
1.3　PLC 前导知识 …… 12
1.3.1　低压电器简介 …… 12
1.3.2　传感器和变送器 …… 13
1.3.3　隔离器 …… 19
1.3.4　浪涌保护器 …… 19
1.3.5　安全栅 …… 20
第 2 章　S7－300/400 PLC 的硬件介绍 …… 22
2.1　S7－300 PLC 概述 …… 22
2.1.1　西门子 PLC 简介 …… 22
2.1.2　S7－300 的基本结构 …… 23
2.2　S7－300 的 CPU 模块及其接线 …… 24
2.2.1　CPU 模块的技术规范 …… 24
2.2.2　紧凑型 CPU 的接线 …… 27
2.3　S7－300 数字量模块及其接线 …… 30
2.3.1　数字量输入模块 SM321 …… 30
2.3.2　数字量输出模块 SM322 …… 31
2.4　S7－300 模拟量模块及接线 …… 34
2.4.1　模拟量输入模块 SM331 连接 …… 34
2.4.2　模拟量输出模块 SM332 连接 …… 37
2.4.3　模拟量输入/输出模块 SM334 连接 …… 37
2.5　S7－300 的其他模块简介 …… 40
2.5.1　S7－300 系列 PLC 的通信处理模块 …… 40
2.5.2　S7－300 系列 PLC 的功能模块 …… 41
2.5.3　ET－200 的模块 …… 41
2.5.4　电源模块 …… 42
2.5.5　接口模块 …… 42
2.5.6　导轨 …… 43
2.6　S7－ 400 PLC 概述 …… 43
2.6.1　S7－400 的结构 …… 43
2.6.2　S7－400 的特点 …… 44
2.7　S7－400 CPU 模块 …… 44
2.7.1　S7－400 CPU 模块的基本结构 …… 45
2.7.2　S7－400 CPU 模块的分类和技术规范 …… 45
2.8　S7－400 PLC 的机架 …… 47
2.8.1　S7－400 PLC 的机架简介 …… 47
2.8.2　UR1 和 UR2 机架（通用机架） …… 48
2.8.3　CR2 和 CR3 机架 …… 48
2.8.4　UR2－H 机架 …… 48
2.9　S7－400 数字量模块及其连接 …… 48

2.9.1 数字量输入模块 SM421 及其连接…… 49
2.9.2 数字量输出模块 SM422 及其连接 …… 50
2.10 S7－400 模拟量模块及其连接 …… 52
2.10.1 模拟量输入模块 SM431 及其连接 …… 52
2.10.2 模拟量输出模块 SM432 及其连接 …… 54
2.11 S7－400 的其他模块 …… 55
2.11.1 电源模块 …… 55
2.11.2 通信处理模块 …… 55
2.11.3 功能模块 …… 55
2.11.4 接口模块 …… 55
**第 3 章 STEP 7 软件使用入门** …… 56
3.1 STEP 7 简介 …… 56
3.1.1 初识 STEP 7 …… 56
3.1.2 安装 STEP 7 的软硬件条件 …… 57
3.1.3 安装 STEP 7 注意事项 …… 57
3.1.4 安装 STEP 7 的过程 …… 57
3.1.5 卸载 STEP 7 的过程 …… 59
3.2 编程界面的 SIMATIC 管理器 … 59
3.2.1 创建项目…… 59
3.2.2 编辑项目…… 60
3.3 硬件组态与参数设置…… 62
3.3.1 硬件组态…… 62
3.3.2 参数设定…… 63
3.3.3 硬件的更新和 GSD 文件安装 … 69
3.4 下载和上传…… 71
3.4.1 下载 …… 71
3.4.2 上传 …… 73
3.5 软件编程…… 75
3.6 打印和归档…… 76
3.6.1 打印 …… 76
3.6.2 归档 …… 76
3.7 用 STEP 7 V5.5 建立一个完整的项目…… 77
3.8 使用帮助…… 79
3.8.1 查找关键字或者功能 …… 79
3.8.2 了解某个逻辑块 FB/SFB/FC/SFC 的功能及引脚的定义 …… 80
**第 4 章 S7－300/400 PLC 的编程语言** …… 81
4.1 S7－300/400 PLC 的编程基础知识…… 81
4.1.1 编程元件…… 81
4.1.2 数制和数据类型 …… 82
4.1.3 寻址方式…… 86
4.1.4 编程语言…… 90
4.2 CPU 中的寄存器 …… 91
4.3 位逻辑指令…… 94
4.4 定时器与计数器指令 …… 102
4.4.1 定时器 …… 102
4.4.2 IEC 定时器 …… 112
4.4.3 计数器 …… 112
4.4.4 IEC 计数器 …… 116
4.5 数据处理与运算指令 …… 117
4.5.1 装载与传送指令…… 117
4.5.2 比较指令 …… 122
4.5.3 转换指令 …… 125
4.5.4 移位与循环指令…… 130
4.5.5 算术运算指令 …… 136
4.5.6 逻辑控制指令 …… 144
4.5.7 字逻辑运算指令…… 146
4.5.8 其他指令 …… 147
4.6 实例 …… 148
4.6.1 电动机的控制 …… 149
4.6.2 定时器和计数器应用 …… 151
**第 5 章 S7－300/400 PLC 的程序结构**…… 154
5.1 STEP 7 编程方法简介 …… 154
5.2 功能、数据块和功能块 …… 154
5.2.1 块的概述 …… 154
5.2.2 功能（FC）及其应用…… 156
5.2.3 共享数据块（DB）及其应用 …… 159
5.2.4 自定义数据类型（UDT）及其

应用 …………………………… 161
5.2.5 功能块（FB）及其应用……… 163
5.2.6 系统功能（SFC）及其应用 … 166
5.3 多重背景 ……………………… 168
5.3.1 多重背景的简介…………… 168
5.3.2 多重背景的应用…………… 168
5.4 库的应用 ……………………… 170
5.4.1 库的介绍 ………………… 170
5.4.2 库的应用介绍 …………… 171
5.5 组织块（OB）及其应用……… 173
5.5.1 中断的概述 ……………… 173
5.5.2 主程序（OB1）…………… 175
5.5.3 日期时钟中断组织块及其
应用 …………………… 175
5.5.4 循环中断组织块及其应用 …… 176
5.5.5 硬件中断组织块及其应用 …… 178
5.5.6 错误处理组织块及其应用 …… 179
5.5.7 背景组织块 ……………… 193
5.5.8 启动组织块及其应用 ……… 194
5.6 实例 ………………………… 195
**第 6 章 S7-300/400 PLC 的编程方法与调试** …………………… 207
6.1 功能图 ……………………… 207
6.1.1 功能图的画法 …………… 207
6.1.2 梯形图编程的原则 ………… 213
6.2 逻辑控制的梯形图编程方法 … 215
6.2.1 经验设计法 ……………… 215
6.2.2 功能图设计法 …………… 216
6.3 S7-300/400 PLC 的调试
方法 …………………………… 226
6.3.1 用变量监控表进行调试 ……… 226
6.3.2 使用 PLCSIM 软件进行调试 … 230
6.3.3 使用“断点”调试程序 ……… 233
6.3.4 使用参考数据调试程序 ……… 234
6.4 实例 ………………………… 237
**第 7 章 西门子 PLC 的 SCL 和 GRAPH 编程** …………………… 243
7.1 西门子 PLC 的 SCL 编程……… 243
7.1.1 S7-SCL 简介……………… 243
7.1.2 S7-SCL 的软件包安装 ……… 243
7.1.3 S7-SCL 源文件编辑器 ……… 244
7.1.4 S7-SCL 编程语言 ………… 245
7.1.5 S7-SCL 应用举例 ………… 259
7.2 西门子 PLC 的 GRAPH 编程 … 262
7.2.1 S7-GRAPH 简介 ………… 262
7.2.2 S7-GRAPH 的安装与
兼容性 ………………… 263
7.2.3 S7-GRAPH 的应用基础……… 263
7.2.4 S7-GRAPH 的应用举例……… 270
**第 8 章 S7-300/400 PLC 的通信及其应用**……………………… 275
8.1 通信基础知识 ………………… 275
8.1.1 通信的基本概念…………… 275
8.1.2 PLC 网络的术语解释 ……… 276
8.1.3 RS-485 标准串行接口 ……… 278
8.1.4 OSI 参考模型……………… 279
8.1.5 SIMATIC NET 工业网络 ……… 280
8.2 现场总线概述 ………………… 281
8.2.1 现场总线的概念…………… 281
8.2.2 主流现场总线的简介 ……… 282
8.2.3 现场总线的特点…………… 282
8.2.4 现场总线的现状…………… 283
8.2.5 现场总线的发展…………… 283
8.3 MPI 通信及其应用 …………… 283
8.3.1 MPI 通信简介……………… 283
8.3.2 S7-200 与 S7-300 间的 MPI
通信 …………………… 284
8.3.3 S7-300 与 S7-300 间的 MPI
通信 …………………… 290
8.3.4 S7-300/400 与 S7-400 间的
MPI 通信 ……………… 297
8.4 PROFIBUS 通信及其应用 …… 300
8.4.1 PROFIBUS 通信概述 ……… 300
8.4.2 PROFIBUS 总线拓扑结构 …… 302
8.4.3 S7-300 与 ET200M 的 PROFIBUS-
DP 通信 ……………… 305
8.4.4 S7-300 与 S7-200 间的 PROFIBUS-
DP 通信………………… 310

8.4.5 S7－300 与 S7－300 间的 PROFIBUS－DP 通信 …… 318
8.4.6 CP342－5 作主站的 PROFIBUS－DP 通信 …… 327
8.4.7 CP342－5 作从站的 PROFIBUS－DP 通信 …… 331
8.4.8 DP/DP Coupler 的 PROFIBUS－DP 通信 …… 337
8.5 以太网通信及其应用 …… 345
8.5.1 以太网通信基础 …… 345
8.5.2 S7－300 间的以太网通信 …… 348
8.5.3 S7－400 与远程 IO 模块 ET200 间的 PROFINET 通信 …… 356
8.5.4 S7－400 与 S7－200 SMART 间的以太网通信 …… 362
8.6 网络路由功能 …… 367
8.6.1 网络路由功能概述 …… 367
8.6.2 网络路由功能应用 …… 367
**第 9 章 S7－300/400 PLC 的 PID 控制技术** …… 373
9.1 PID 控制简介 …… 373
9.1.1 PID 控制原理简介 …… 373
9.1.2 PID 控制的算法和图解 …… 374
9.1.3 PID 控制器的参数整定 …… 377
9.2 利用 S7－300 进行电炉的温度控制 …… 379
**第 10 章 西门子人机界面（HMI）应用** …… 388
10.1 人机界面简介 …… 388
10.1.1 认识人机界面 …… 388
10.1.2 触摸屏的通信连接 …… 388
10.2 WinCC flexible 软件简介 …… 389
10.2.1 认识 WinCC flexible 软件 …… 389
10.2.2 安装 WinCC flexible 软件 …… 390
10.3 变量组态 …… 390
10.3.1 变量类型 …… 390
10.3.2 函数 …… 391
10.4 画面组态 …… 392
10.4.1 按钮组态 …… 392
10.4.2 IO 域组态 …… 393
10.4.3 开关组态 …… 394
10.4.4 图形输入/输出对象组态 …… 396
10.4.5 时钟和日期的组态 …… 398
10.4.6 间接寻址和符号 IO 域组态 …… 399
10.4.7 图形 IO 域组态 …… 401
10.4.8 画面的切换 …… 402
10.5 用户管理 …… 404
10.5.1 用户管理的基本概念 …… 404
10.5.2 用户管理编辑器 …… 404
10.5.3 访问保护 …… 405
10.6 创建一个简单的触摸屏的工程 …… 406
**第 11 章 S7－300 PLC 在变频调速系统中的应用** …… 414
11.1 变频器的概述 …… 414
11.1.1 变频器的发展历史 …… 414
11.1.2 变频器的分类 …… 417
11.2 西门子 MM440 变频器使用简介 …… 418
11.2.1 认识变频器 …… 418
11.2.2 西门子低压变频器简介 …… 420
11.2.3 西门子 MM440 变频器使用简介 …… 421
11.3 变频器多段频率给定 …… 424
11.4 变频器的模拟量频率给定 …… 429
11.5 变频器的通信频率给定 …… 433
11.5.1 MM440 变频器通信的基本知识 …… 433
11.5.2 实例 …… 436
11.5.3 S7－300 通过 PROFIBUS 现场总线修改 MM440 变频器的参数 …… 443
11.5.4 S7－300 通过 PROFIBUS 现场总线控制 S120 …… 449
11.6 变频器的调试 …… 458
11.6.1 STARTER 软件与传动装置常用通信连接方式 …… 458
11.6.2 用 STARTER 软件调试 MM440 实例 …… 459

第 12 章　西门子 S7－300/400 PLC 的故障诊断技术 ………………… 469
12.1　PLC 控制系统的故障诊断概述…………………………… 469
12.1.1　引发 PLC 故障的外部因素…… 469
12.1.2　PLC 的故障类型和故障信息 ……………………… 470
12.1.3　PLC 故障诊断方法 ………… 471
12.1.4　PLC 外部故障诊断方法……… 471
12.2　S7－300/400 故障诊断技术…………………………… 475
12.2.1　使用状态和出错 LED 进行故障诊断 …………………… 475
12.2.2　用 STEP 7 诊断视图进行故障诊断 …………………… 478
12.2.3　用通信块（如 SFB15）的输出参数/返回值（RET_VAL）诊断故障 …………………… 486
12.2.4　用组织块进行故障诊断……… 488
12.2.5　用 SFC13/SFC51/FC10/FB10/FC3 进行故障诊断 ………… 497
12.2.6　用 BT200 PROFIBUS 总线物理测试工具诊断故障 ………… 514
12.2.7　用 PC 网卡/CP443－1/CP343－1 等通信处理器诊断以太网故障 ……………………… 517
12.2.8　用 CP5611/CP5511/CP342－5 等通信处理器诊断通信故障 …… 520
12.2.9　用软件 Amprolyzer 诊断故障 ……………………… 526
12.3　各种故障诊断的比较………… 529
12.4　故障诊断的显示……………… 530
12.4.1　位发送故障显示 …………… 530
12.4.2　消息号发送故障显示 ……… 532
12.4.3　用 S7－PDIAG 软件组态过程诊断 ……………………… 540
12.5　工程常见调试和故障诊断 30 例 ………………………… 547
第 13 章　S7－300/400 PLC 工程应用 ………………………… 557
13.1　啤酒灌装线系统的 PLC 控制…………………………… 557
13.1.1　系统软硬件配置 …………… 557
13.1.2　编写程序 …………………… 558
13.2　干粉喷射系统的 PLC 控制 … 564
13.2.1　硬件系统集成 ……………… 564
13.2.2　编写程序 …………………… 572
13.2.3　完成触摸屏 HMI 项目 ……… 593
参考文献………………………………… 596

# 第 1 章　可编程序控制器（PLC）基础

本章介绍可编程序控制器的历史、功能、特点、应用范围、发展趋势、在我国的使用情况、结构和工作原理等知识，使读者初步了解可编程序控制器，这是学习本书后续内容的必要准备。

## 1.1　概述

可编程序控制器（Programmable Logic Controller）简称 PLC，国际电工委员会（IEC）于 1985 年对可编程序控制器作了如下定义：可编程序控制器是一种数字运算操作的电子系统，专为在工业环境下应用而设计。它采用可编程序的存储器，用来在其内部存储执行逻辑运算、顺序控制、定时、计数和算术运算等操作的指令，并通过数字、模拟的输入和输出，控制各种类型的机械或生产过程。可编程序控制器及其有关设备，都应按易于与工业控制系统连成一个整体，易于扩充功能的原则设计。PLC 是一种工业计算机，其种类繁多，不同厂家的产品有各自的特点，但作为工业标准设备，可编程序控制器又有一定的共性。

### 1.1.1　PLC 的发展历史

20 世纪 60 年代以前，汽车生产线的自动控制系统基本上都是由继电器控制装置构成的。当时每次改型都直接导致继电器控制装置的重新设计和安装，福特汽车公司的老板曾经说：“无论顾客需要什么样的汽车，福特的汽车永远是黑色的”。从侧面反映汽车改型和升级换代比较困难。为了改变这一现状，1969 年，美国的通用汽车公司（GM）公开招标，要求用新的装置取代继电器控制装置，并提出十项招标指标，要求编程方便、现场可修改程序、维修方便、采用模块化设计、体积小以及可与计算机通信等。同一年，美国数字设备公司（DEC）研制出了世界上第一台可编程序控制器 PDP－14，在美国通用汽车公司的生产线上试用成功，并取得了满意的效果，可编程序控制器从此诞生。由于当时的 PLC 只能取代继电器接触器控制，功能仅限于逻辑运算、计时及计数等，所以称为“可编程逻辑控制器”。伴随着微电子技术、控制技术与信息技术的不断发展，可编程序控制器的功能不断增强。美国电气制造商协会（NEMA）于 1980 年正式将其命名为“可编程序控制器”，简称 PC，由于这个名称和个人计算机的简称相同，容易混淆，因此在我国，很多人习惯称可编程序控制器为 PLC。

由于 PLC 具有易学易用、操作方便、可靠性高、体积小、通用灵活和使用寿命长等一系列优点，因此，很快就在工业中得到了广泛应用。同时，这一新技术也受到其他国家的重视。1971 年日本引进这项技术，很快研制出日本第一台 PLC；欧洲于 1973 年研制出第一台 PLC；我国从 1974 年开始研制，1977 年国产 PLC 正式投入工业应用。

进入 20 世纪 80 年代以来，随着电子技术的迅猛发展，以 16 位和 32 位微处理器构成的微机化 PLC 得到快速发展，（例如 GE 的 RX7i，使用的是赛扬 CPU，其主频达 1 GHz，其信

息处理能力几乎和个人电脑相当）使得 PLC 在设计、性能价格比以及应用方面有了突破，不仅控制功能增强、功耗和体积减小、成本下降、可靠性提高以及编程和故障检测更为灵活方便，而且随着远程 I/O 和通信网络、数据处理和图像显示的发展，PLC 普遍应用于控制复杂的生产过程。PLC 已经成为工厂自动化的三大支柱之一。

### 1.1.2 PLC 的主要特点

PLC 之所以高速发展，除了工业自动化的客观需要外，还有许多适合工业控制的独特优点，它较好地解决了工业控制领域中普遍关心的可靠、安全、灵活、方便以及经济等问题，其主要特点如下：

**1. 抗干扰能力强，可靠性高**

在传统的继电器控制系统中，使用了大量的中间继电器、时间继电器，由于器件的固有缺点，如器件老化、接触不良及触点抖动等现象，大大降低了系统的可靠性。而在 PLC 控制系统中大量的开关动作由无触点的半导体电路完成，因此故障大大减少。

此外，PLC 的硬件和软件方面采取了措施，提高了其可靠性。在硬件方面，所有的 I/O 接口都采用了光电隔离，使得外部电路与 PLC 内部电路实现了物理隔离。各模块均采用屏蔽措施，以防止辐射干扰。电路中采用了滤波技术，以防止或抑制高频干扰。在软件方面，PLC 具有良好的自诊断功能，一旦系统的软硬件发生异常情况，CPU 会立即采取有效措施，以防止故障扩大。通常 PLC 具有看门狗功能。

对于大型的 PLC 系统，还可以采用双 CPU 构成冗余系统或者三 CPU 构成表决系统，使系统的可靠性进一步提高。

**2. 程序简单易学，系统的设计调试周期短**

PLC 是面向用户的设备。PLC 的生产厂家充分考虑到现场技术人员的技能和习惯，可采用梯形图或面向工业控制的简单指令形式。梯形图与继电器原理图很相似，直观、易懂、易掌握，不需要学习专门的计算机知识和语言。设计人员可以在设计室设计、修改和模拟调试程序，非常方便。

**3. 安装简单，维修方便**

PLC 不需要专门的机房，可以在各种工业环境下直接运行，使用时只需将现场的各种设备与 PLC 相应的 I/O 端相连接，即可投入运行。各种模块上均有运行和故障指示装置，便于用户了解运行情况和查找故障。

**4. 采用模块化结构，体积小，重量轻**

为了适应工业控制需求，除整体式 PLC 外，绝大多数 PLC 采用模块化结构。PLC 的各部件，包括 CPU、电源以及 I/O 等都采用模块化设计。此外，PLC 相对于通用工控机，其体积和重量要小得多。

**5. 丰富的 I/O 接口模块，扩展能力强**

PLC 针对不同的工业现场信号（如交流或直流、开关量或模拟量、电压或电流、脉冲或电位以及强电或弱电等）有相应的 I/O 模块与工业现场的器件或设备（如按钮、行程开关、接近开关、传感器及变送器、电磁线圈及控制阀等）直接连接。另外，为了提高操作性能，有多种人 – 机对话的接口模块；为了组成工业局部网络，有多种通信联网的接口模块等。

### 1.1.3　PLC 的应用范围

目前，PLC 在国内外已广泛应用于专用机床、机床、控制系统、自动化楼宇、钢铁、石油、化工、电力、建材、汽车、纺织机械、交通运输、环保以及文化娱乐等各行各业。随着 PLC 性能价格比的不断提高，其应用范围还将不断扩大，其应用场合可以说是无处不在，具体应用大致可归纳为如下几类：

**1. 顺序控制**

顺序控制是 PLC 最基本、最广泛应用的领域，它取代传统的继电器顺序控制，PLC 用于单机控制、多机群控制及自动化生产线的控制。例如数控机床、注塑机、印刷机械、电梯控制和纺织机械等。

**2. 计数和定时控制**

PLC 为用户提供了足够的定时器和计数器，并设置相关的定时和计数指令，PLC 的计数器和定时器准确度高、使用方便，可以取代继电器系统中的时间继电器和计数器。

**3. 位置控制**

目前大多数的 PLC 制造商都提供拖动步进电动机或伺服电动机的单轴或多轴位置控制模块，这一功能可广泛用于各种机械，如金属切削机床、装配机械等。

**4. 模拟量处理**

PLC 通过模拟量的输入/输出模块，实现模拟量与数字量的转换，并对模拟量进行控制，有的还具有 PID 控制功能。例如用于锅炉的水位、压力和温度控制。

**5. 数据处理**

现代的 PLC 具有数学运算、数据传递、转换、排序和查表等功能，也能完成数据的采集、分析和处理。

**6. 通信联网**

PLC 的通信包括 PLC 相互之间、PLC 与上位计算机以及 PLC 和其他智能设备之间的通信。PLC 系统与通用计算机可以直接或通过通信处理单元、通信转接器相连构成网络，以实现信息的交换，并可构成“集中管理、分散控制”的分布式控制系统，满足工厂自动化系统的需要。

### 1.1.4　PLC 的分类与性能指标

**1. PLC 的分类**

（1）从组成结构形式分类

可以将 PLC 分为两类：一类是整体式 PLC（也称单元式），其特点是电源、中央处理单元和 I/O 接口都集成在一个机壳内；另一类是标准模板式结构化的 PLC（也称组合式），其特点是电源模板、中央处理单元模板和 I/O 模板等在结构上是相互独立的，可根据具体的应用要求，选择合适的模块，安装在固定的机架或导轨上，构成一个完整的 PLC 应用系统。

（2）按 I/O 点容量分类

1）小型 PLC。小型 PLC 的 I/O 点数一般在 128 点以下。

2）中型 PLC。中型 PLC 采用模块化结构，其 I/O 点数一般在 256～1024 点之间。

3）大型 PLC。一般 I/O 点数在 1024 点以上的称为大型 PLC。

**2. PLC 的性能指标**

各厂家的 PLC 虽然各有特色，但其主要性能指标是相同的。

（1）输入/输出（I/O）点数

输入/输出（I/O）点数是最重要的一项技术指标，是指 PLC 面板上连接外部输入、输出的端子数，常称为“点数”，用输入与输出点数的和表示。点数越多表示 PLC 可接入的输入器件和输出器件越多，控制规模越大。点数是 PLC 选型时最重要的指标之一。

（2）扫描速度

扫描速度是指 PLC 执行程序的速度。以 ms/K 为单位，即执行 1 K 步指令所需的时间。1 步占 1 个地址单元。

（3）存储容量

存储容量通常用 K 字（KW）或 K 字节（KB）、K 位来表示。这里 1 K = 1024。有的 PLC 用“步”来衡量，一步占用一个地址单元。存储容量表示 PLC 能存放多少用户程序。例如，三菱型号为 FX2N－48MR 的 PLC 存储容量为 8000 步。有的 PLC 的存储容量可以根据需要配置，有的 PLC 的存储器可以扩展。

（4）指令系统

指令系统表示该 PLC 软件功能的强弱。指令越多，编程功能就越强。

（5）内部寄存器（继电器）

PLC 内部有许多寄存器用来存放变量、中间结果、数据等，还有许多辅助寄存器可供用户使用。因此寄存器的配置也是衡量 PLC 功能的一项指标。

（6）扩展能力

扩展能力是反映 PLC 性能的重要指标之一。PLC 除了主控模块外，还可配置实现各种特殊功能的高功能模块。例如 A－D 模块、D－A 模块、高速计数模块及远程通信模块等。

### 1.1.5 PLC 与继电器系统的比较

在 PLC 出现以前，继电器硬接线电路是逻辑、顺序控制的唯一执行者，它结构简单、价格低廉，一直被广泛应用。PLC 出现后，几乎所有的方面都超过继电器控制系统，两者的性能比较见表 1–1。

**表 1–1 可编程序控制器与继电器控制系统的比较**

| 序 号 | 比较项目 | 继电器控制 | 可编程序控制器控制 |
|---|---|---|---|
| 1 | 控制逻辑 | 硬接线多、体积大、连线多 | 软逻辑、体积小、接线少、控制灵活 |
| 2 | 控制速度 | 通过触点开关实现控制，动作受继电器硬件限制，通常超过 10 ms | 由半导体电路实现控制，指令执行时间段，一般为微秒级 |
| 3 | 定时控制 | 由时间继电器控制，准确度差 | 由集成电路的定时器完成，准确度高 |
| 4 | 设计与施工 | 设计、施工及调试必须按照顺序进行，周期长 | 系统设计完成后，施工与程序设计同时进行，周期短 |
| 5 | 可靠性与维护 | 继电器的触点寿命短，可靠性和维护性差 | 无触点、寿命长、可靠性高，有自诊断功能 |
| 6 | 价格 | 价格低 | 价格高 |

### 1.1.6 PLC 与微机的比较

采用微电子技术制造的 PLC 与微机一样，也由 CPU、ROM（或者 FLASH）、RAM 以及

I/O 接口等组成，但又不同于一般的微机，PLC 采用了特殊的抗干扰技术，是一种特殊的工业控制计算机，更加适合工业控制。两者的性能比较见表 1-2。

**表 1-2　PLC 与微机的比较**

| 序　　号 | 比 较 项 目 | PLC 控制 | 微 机 控 制 |
|---|---|---|---|
| 1 | 应用范围 | 工业控制 | 科学计算、数据处理、计算机通信 |
| 2 | 使用环境 | 工业现场 | 具有一定温度和湿度的机房 |
| 3 | 输入/输出 | 控制强电设备，需要隔离 | 与主机弱电联系，不隔离 |
| 4 | 程序设计 | 一般使用梯形图语言，易学易用 | 编程语言丰富，如 C、BASIC 等 |
| 5 | 系统功能 | 自诊断、监控 | 使用操作系统 |
| 6 | 工作方式 | 循环扫描方式和中断方式 | 中断方式 |

## 1.1.7　PLC 的发展趋势

PLC 的发展趋势主要有以下几个方面：

1）向高性能、高速度、大容量发展。

2）网络化。强化通信能力和网络化，向下将多个 PLC 或者多个 I/O 框架相连；向上与工业计算机、以太网等相连，构成整个工厂的自动化控制系统。即便是微型的 S7－200 系列 PLC 也能组成多种网络，通信功能十分强大。

3）小型化、低成本、简单易用。目前，有的小型 PLC 的价格只需几百元人民币。

4）不断提高编程软件的功能。编程软件可以对 PLC 控制系统的硬件组态，在屏幕上可以直接生成和编辑梯形图、指令表、功能块图和顺序功能图程序，并可以实现不同编程语言的相互转换。程序可以下载、存盘和打印，通过网络或电话线，还可以实现远程编程。

5）适合 PLC 应用的新模块。随着科技的发展，对工业控制领域将提出更高的、更特殊的要求，因此，必须开发特殊功能模块来满足这些要求。

6）PLC 的软件化与 PC 化。目前已有多家厂商推出了在 PC 上运行的可实现 PLC 功能的软件包，也称为“软 PLC”，“软 PLC”的性能价格比比传统的“硬 PLC”更高，是 PLC 的一个发展方向。PC 化的 PLC 类似于 PLC，但它采用了 PC 的 CPU，功能十分强大，如 GE 的 Rx7i 和 Rx3i 使用的就是工控机用的赛扬 CPU，主频已经达到 1 GHz。

## 1.1.8　PLC 在我国

**1. 国外 PLC 品牌**

目前 PLC 在我国得到了广泛的应用，很多知名厂家的 PLC 在我国都有应用。

1）美国是 PLC 生产大国，有 100 多家 PLC 生产厂家。其中 A－B 公司的 PLC 产品规格比较齐全，主推大中型 PLC，主要产品系列是 PLC－5。通用电气也是知名 PLC 生产厂商，大中型 PLC 产品系列有 RX3i 和 RX7i 等。德州仪器也生产大、中、小全系列 PLC 产品。

2）欧洲的 PLC 产品也久负盛名。德国的西门子公司、AEG 公司和法国的 TE（施耐德）公司都是欧洲著名的 PLC 制造商。其中西门子公司的 PLC 产品与美国的 A－B 的 PLC 产品齐名。

3）日本的小型 PLC 具有一定的特色，性价比高，知名的品牌有三菱、欧姆龙、松下、

富士、日立和东芝等，在小型机市场，日系 PLC 的市场份额曾经高达70%。

**2. 国产 PLC 品牌**

我国自主品牌的 PLC 生产厂家有近 30 余家。在目前已经上市的众多 PLC 产品中，还没有形成规模化的生产和名牌产品，甚至还有一部分是以仿制、来件组装或“贴牌”方式生产。单从技术角度来看，国产小型 PLC 与国际知名品牌小型 PLC 差距正在缩小，使用越来越多。例如和利时、深圳汇川和无锡信捷等公司生产的微型 PLC 已经比较成熟，其可靠性在许多低端应用中得到了验证，逐渐被用户认可，但其知名度与世界先进水平还有一定的差距。

总的来说，我国使用的小型 PLC 主要以日本和国产品牌为主，而大中型 PLC 主要以欧美品牌为主。目前95%以上的 PLC 市场被国外品牌所占领。

## 1.2　PLC 的结构和工作原理

### 1.2.1　PLC 的硬件组成

PLC 种类繁多，但其基本结构和工作原理相同。PLC 的功能结构区由 CPU（中央处理器）、存储器和输入/输出接口三部分组成，如图 1-1 所示。

**1. CPU（中央处理器）**

CPU 的功能是完成 PLC 内所有的控制和监视操作。中央处理器一般由控制器、运算器和寄存器组成。CPU 通过数据总线、地址总线和控制总线与存储器、输入/输出接口电路连接。

**2. 存储器**

在 PLC 中使用两种类型的存储器：一种是只读类型的存储器，如 EPROM 和 EEPROM，另一种是可读/写的随机存储器 RAM。PLC 的存储器分为 5 个区域，如图 1-2 所示。

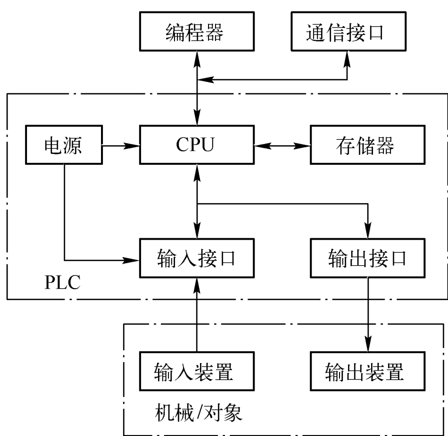



图 1-1　PLC 结构框图

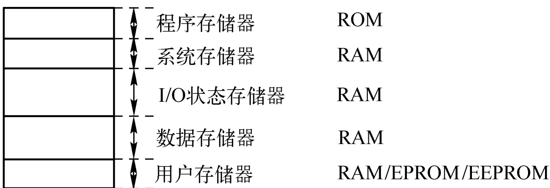



图 1-2　存储器的区域划分

程序存储器的类型是只读存储器（ROM），PLC 的操作系统存放在这里，程序由制造商固化，通常不能修改。存储器中的程序负责解释和编译用户编写的程序、监控 I/O 口的状

态、对 PLC 进行自诊断以及扫描 PLC 中的程序等。系统存储器属于随机存储器（RAM），主要用于存储中间计算结果和数据、系统管理，有的 PLC 厂家用系统存储器存储一些系统信息如错误代码等，系统存储器不对用户开放。I/O 状态存储器属于随机存储器，用于存储 I/O 装置的状态信息，每个输入模块和输出模块都在 I/O 映像表中分配一个地址，而且这个地址是唯一的。数据存储器属于随机存储器，主要用于数据处理功能，为计数器、定时器、算术计算和过程参数提供数据存储。有的厂家将数据存储器细分为固定数据存储器和可变数据存储器。用户编程存储器，其类型可以是随机存储器、可擦除存储器（EPROM）和电擦除存储器（EEPROM），高档的 PLC 还可以用 FLASH。用户编程存储器主要用于存放用户编写的程序。存储器的关系如图 1-3 所示。

只读存储器可以用来存放系统程序，PLC 断电后再上电，系统内容不变且重新执行。只读存储器也可用来固化用户程序和一些重要参数，以免因偶然操作失误而造成程序和数据的破坏或丢失。随机存储器中一般存放用户程序和系统参数。当 PLC 处于编程工作时，CPU 从 RAM 中取指令并执行。用户程序执行过程中产生的中间结果也在 RAM 中暂时存放。RAM 通常由 CMOS 型集成电路组成，功耗小，但断电时内容消失，所以一般使用大电容或后备锂电池保证掉电后 PLC 的内容在一定时间内不丢失。

**3. 输入/输出接口**

PLC 的输入和输出信号可以是开关量或模拟量。输入/输出接口是 PLC 内部弱电（low power）信号和工业现场强电（high power）信号联系的桥梁。输入/输出接口主要有两个作用，一是利用内部的电隔离电路将工业现场和 PLC 内部进行隔离，起保护作用；二是调理信号，可以把不同的信号（如强电、弱电信号）调理成 CPU 可以处理的信号（5 V、3. 3 V 或 2. 7 V 等），如图 1-4 所示。

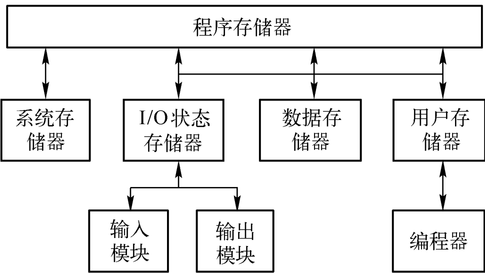



图 1-3　存储器的关系

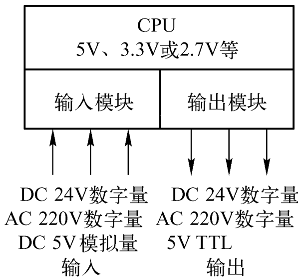



图 1-4　输入/输出接口

输入/输出接口模块是 PLC 系统中最大的部分，输入/输出接口模块通常需要电源，输入电路的电源可以由外部提供，对于模块化的 PLC 还需要背板（安装机架）。

（1）输入接口电路

1）输入接口电路的组成和作用。输入接口电路由接线端子、输入调理和电平转换电路、模块状态显示、电隔离电路和多路选择开关模块组成，如图 1-5 所示。现场的信号必须连接在输入端子才可能将信号输入到 CPU 中，它提供了外部信号输入的物理接口；调理和电平转换电路十分重要，可以将工业现场的信号（如强电 AC 220 V 信号）转化成电信号（CPU 可以识别的弱电信号）；电隔离电路主要是利用电隔离器件将工业现场的机械或者电输入信号和 PLC 的 CPU 的信号隔开，它能确保过高的电干扰信号和浪涌不串入 PLC 的微处

理器，起保护作用。有三种隔离方式，用得最多的是光电隔离，其次是变压器隔离和干簧继电器隔离；当外部有信号输入时，输入模块上有指示灯显示，这个电路比较简单，当线路中有故障时，它帮助用户查找故障，由于氖灯或 LED 灯的寿命比较长，所以这个灯通常是氖灯或 LED 灯；多路选择开关接受调理完成的输入信号，并存储在多路开关模块中，当输入循环扫描时，多路开关模块中信号输送到 I/O 状态寄存器中。

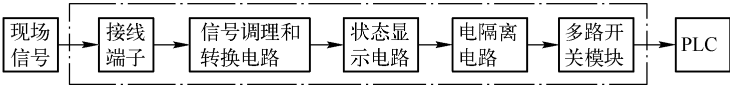



图 1-5　输入接口的结构

2）输入信号的设备的种类。输入信号可以是离散信号和模拟信号。当输入端是离散信号时，输入端的设备类型可以是限位开关、按钮、压力继电器、继电器触点、接近开关、选择开关及光电开关等，如图 1-6 所示。当输入为模拟量输入时，输入设备的类型可以是压力传感器、温度传感器、流量传感器、电压传感器、电流传感器及力传感器等。

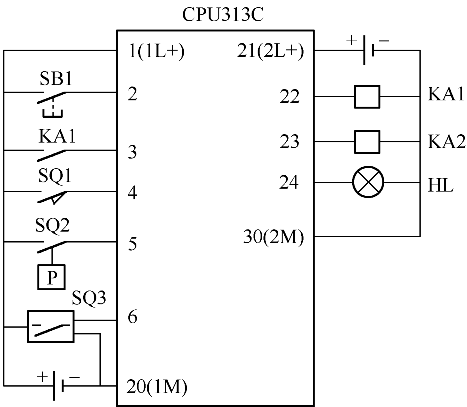



图 1-6　输入/输出接口

（2）输出接口电路

1）输出接口电路的组成和作用。输出接口电路由多路选择开关模块、信号锁存器、电隔离电路、模块状态显示、输出电平转换电路和接线端子组成，如图 1-7 所示。在输出扫描期间，多路选择开关模块接受来自映像表中的输出信号，并对这个信号的状态和目标地址进行译码，最后将信息送给锁存器；信号锁存器是将多路选择开关模块的信号保存起来，直到下一次更新；输出接口的电隔离电路作用和输入模块的一样，但是由于输出模块输出的信号比输入信号要强得多，因此要求隔离电磁干扰和浪涌的能力更高；输出电平转换电路将隔离电路送来的信号放大成可以足够驱动现场设备的信号，放大器件可以是双向晶闸管、晶体管和干簧继电器等；输出的接线端子用于将输出模块与现场设备相连接。

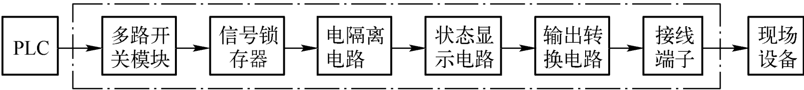



图 1-7　输出接口的结构

PLC 有三种输出接口形式，继电器输出、晶体管输出和晶闸管输出形式。继电器输出形式的 PLC 的负载电源可以是直流电源或交流电源，但其输出响应频率较慢，其内部电路如图 1-8 所示。晶体管输出的 PLC 负载电源是直流电源，其输出响应频率较快，其内部电路如图 1-9 所示。晶闸管输出形式的 PLC 的负载电源是交流电源，西门子 S7－200 系列 PLC 的 CPU 模块暂时还没有晶闸管输出形式的产品出售，但三菱 FX 系列有这种产品。选型时要

特别注意 PLC 的输出形式。

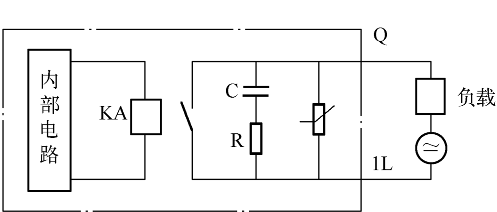



图 1-8　继电器输出内部电路

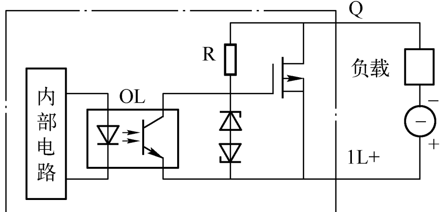



图 1-9　晶体管输出内部电路

2）输出信号的设备的种类。输出信号可以是离散信号和模拟信号。当输出端是离散信号时，输出端的设备类型可以是电磁阀的线圈、电动机起动器、控制柜的指示器、接触器线圈、LED 灯、指示灯、继电器线圈、报警器和蜂鸣器等。当输出为模拟量输出时，输出设备的类型可以是流量阀、AC 驱动器（如交流伺服驱动器）、DC 驱动器、模拟量仪表、温度控制器和流量控制器等。

**【关键点】** PLC 的继电器型输出虽然响应速度慢，但其驱动能力强，一般为 2 A，这是继电器型输出 PLC 的一个重要的优点。一些特殊性号的 PLC，如西门子 LOGO！的某些型号驱动能力可达 5 A 和 10 A，能直接驱动接触器。此外，从图 1-8 中可以看出继电器型输出形式的 PLC，对于一般的误接线，通常不会引起 PLC 内部器件的烧毁（高于交流 220 V 电压是不允许的）。因此，继电器输出形式是选型时的首选，在工程实践中，用得比较多。

晶体管输出的 PLC 的输出电流一般小于 1 A，西门子 S7－200 的输出电流源是 0.75 A（西门子有的型号的 PLC 的输出电流甚至只有 0.5 A），可见晶体管输出的驱动能力较小。此外，图 1-9 可以看出晶体管型输出形式的 PLC，对于一般的误接线，可能会引起 PLC 内部器件的烧毁，所以要特别注意。

**【例 1-1】** 某学生按图 1-10 所示接线，之后按下 SB1、SB2 和 SB3 按钮，发现输入端的指示灯没有显示，PLC 中没有程序，但灯 HL 常亮，接线没有错误，＋24 V 电源也正常。学生的分析是输入和输出接口烧毁，请问该分析是否正确。

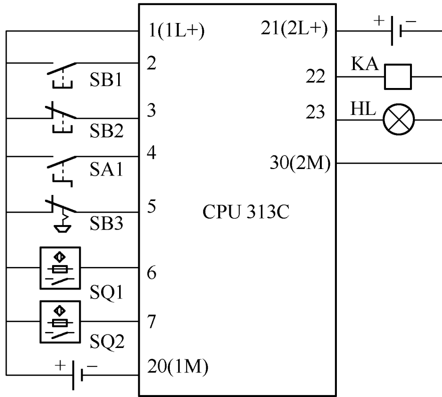



图 1-10　接线图

**解**：分析如下：

1）一般在实验室环境中，输入端口不会烧毁，因为输入接口电路有光电隔离电路保护，除非有较大电压（如交流 220 V）的误接入，而且烧毁输入接口一般也不会所有的接口同时烧毁。经过检查，发现接线端子 1 M 是“虚接”，压紧此接线端子后，输入端恢复正常。

2）误接线容易造成晶体管输出回路的器件烧毁，晶体管的击穿会造成回路导通，从而造成 HL 灯常量。

**【关键点】** 本书中所有的 PNP 输入和 NPN 输入都是以传感器为对象，有的资料以 PLC

为对象，则变成 NPN 输入和 PNP 输入，请读者注意。

### 1.2.2 PLC 的工作模式

**1. S7 – 300/400 的工作模式**

S7 – 300/400 共有 4 种工作模式：RUN、RUN – P、STOP 和 MRES，可以通过面板上的模式选择开关来选择。RUN – P 在早期的 CPU 和仿真器（早期和目前的都有）上才有。

① RUN：运行模式。在此模式下，CPU 执行用户程序，可以通过编程器读出或者监控用户程序，但不能修改。

② RUN – P：可编程模式。在此模式下，CPU 执行用户程序，可以通过编程器读出、监控和修改用户程序。新型号的 S7 – 300 没有此模式，但仿真器仍然有此模式。

③ STOP：停止模式。在此模式下，CPU 不执行用户程序，可以通过编程器读出和修改用户程序。

④ MERS：存储器的复位模式。不能保持，将钥匙开关从 STOP 状态扳到 MRES 位置，可复位存储器，使 CPU 回到初始状态。

复位存储器操作：通电后，将开关从 STOP 位置扳到 MRES 位置，“STOP” LED 熄灭 1 s，亮 1 s，再熄灭 1 s 后保持亮。放开开关，使它回到 STOP 位置，然后又回到 MRES，“STOP” LED 以 2 Hz 的频率至少闪动 3 s，表示正在执行复位，最后“STOP” LED 一直亮。

**2. S7 – 300/400 的起动方式**

（1）暖起动（Warm Restart）

S7 – 300 CPU（不包括 CPU 318）只有暖起动。清除过程映像数据以及非保持的 M/T/C 存储器。有保持功能的 M/T/C/DB 存储器将保留原数值。CPU 的暖起动应用最广泛，是默认的起动方式，暖起动时调用 OB100 组织块一次。

（2）热起动（Hot Restart）

在 RUN 状态时，如果电源突然丢失，然后又重新上电，从上次 RUN 模式结束时程序被中断之处继续执行，不对计数器等复位。仅 S7 – 400 有这种模式，且只有带后备电池才能实现。热起动时调用 OB101 组织块一次。

（3）冷起动（Cold Restart）

冷起动时，过程数据区的 I、Q、M、T、C 和 DB 等被复位为零。只有部分 S7 – 400 有此功能，如 CPU 417 和 CPU 417H。冷起动时调用 OB102 组织块一次。

CPU 的起动方式的选择在硬件组态时设定，将在后续章节讲解。

### 1.2.3 PLC 的工作原理

PLC 是一种存储程序的控制器。用户根据某一对象的具体控制要求，编制好控制程序后，用编程器将程序输入到 PLC（或用计算机下载到 PLC）的用户程序存储器中寄存。PLC 的控制功能就是通过运行用户程序来实现的。

PLC 运行程序的方式与微型计算机相比有较大的不同。微型计算机运行程序时，一旦执行到 END 指令，程序运行便结束；而 PLC 从 0 号存储地址所存放的第一条用户程序开始，在无中断或跳转的情况下，按存储地址号递增的方向顺序逐条执行用户程序，直到 END 指令结束。然后再从头开始执行，并周而复始地重复，直到停机或从运行（RUN）切换到停

止（STOP）工作状态。把 PLC 这种执行程序的方式称为扫描工作方式。每扫描完一次程序就构成一个扫描周期。另外，PLC 对输入、输出信号的处理与微型计算机不同。微型计算机对输入、输出信号实时处理，而 PLC 对输入、输出信号是集中批处理。下面具体介绍 PLC 的扫描工作过程。其运行和信号处理示意如图 1-11 所示。

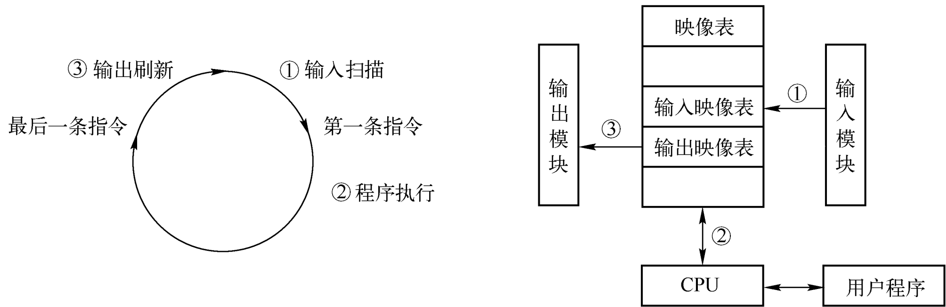



图 1-11　PLC 内部运行和信号处理示意图

PLC 扫描工作方式主要分为三个阶段：输入扫描、程序执行和输出刷新。

**1. 输入扫描**

PLC 在开始执行程序之前，首先扫描输入端子，按顺序将所有输入信号，读入到寄存器-输入状态的输入映像寄存器中，这个过程称为输入扫描。PLC 在运行程序时，所需的输入信号不是现时取输入端子上的信息，而是取输入映像寄存器中的信息。在本工作周期内这个采样结果的内容不会改变，只有到下一个扫描周期输入扫描阶段才被刷新。PLC 的扫描速度很快，取决于 CPU 的时钟速度。

**2. 程序执行**

PLC 完成了输入扫描工作后，按顺序从 0 号地址开始的程序进行逐条扫描执行，并分别从输入映像寄存器、输出映像寄存器以及辅助继电器中获得所需的数据进行运算处理。再将程序执行的结果写入输出映像寄存器中保存。但这个结果在全部程序未被执行完毕之前不会送到输出端子上，也就是物理输出是不会改变的。扫描时间取决于程序的长度、复杂程度和 CPU 的功能。

**3. 输出刷新**

在执行到 END 指令，即执行完用户所有程序后，PLC 上将输出映像寄存器中的内容送到输出锁存器中进行输出，驱动用户设备。扫描时间取决于输出模块的数量。

从以上的介绍可以知道，PLC 程序扫描特性决定了 PLC 的输入和输出状态并不能在扫描的同时改变，例如一个按钮开关的输入信号的输入刚好在输入扫描之后，那么这个信号只有在下一个扫描周期才能被读入。

上述三个步骤是 PLC 的软件处理过程，可以认为就是程序扫描时间。扫描时间通常由三个因素决定，一是 CPU 的时钟速度，越高档的 CPU，时钟速度越高，扫描时间越短；二是 I/O 模块的数量，模块数量越少，扫描时间越短；三是程序的长度，程序长度越短，扫描时间越短。一般的 PLC 执行容量为 1 K 的程序约需要的扫描时间是 1 ~ 10 ms。

图 1-12 所示表达了 PLC 循环扫描工作过程

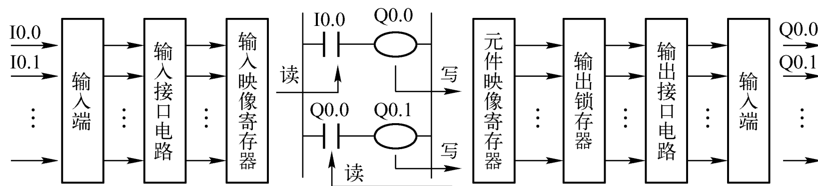



图 1-12　PLC 循环扫描工作过程

### 1.2.4　PLC 的立即输入、输出功能

比较高档的 PLC 都有立即输入、输出功能。

**1. 立即输出功能**

所谓立即输出功能就是输出模块在处理用户程序时，能立即被刷新。PLC 临时挂起（中断）正常运行的程序，将输出映像表中的信息输送到输出模块，立即进行输出刷新，然后再回到程序中继续运行，立即输出的示意图如图 1-13 所示。注意，立即输出功能并不能立即刷新所有的输出模块。

**2. 立即输入功能**

立即输入适用于要求对反映速度很严格的场合，例如几毫秒的时间对于控制来说十分关键的情况下。立即输入时，PLC 立即挂起正在执行的程序，扫描输入模块，然后更新特定的输入状态到输入映像表，最后继续执行剩余的程序，立即输入的示意图如图 1-14 所示。

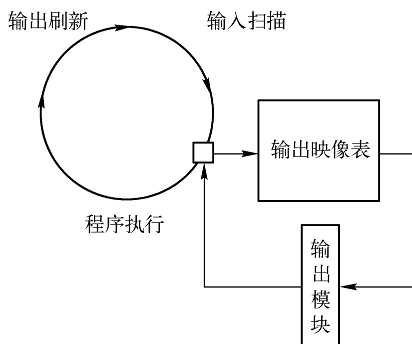



图 1-13　立即输出过程

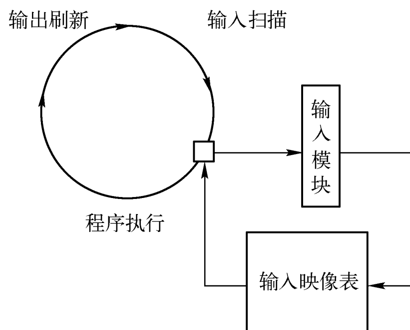



图 1-14　立即输入过程

## 1.3　PLC 前导知识

PLC 控制系统不是孤立的，必然需要使用各种输入、输出、转换和保护等外围器件，以下将对常用的外围器件进行简介。

### 1.3.1　低压电器简介

一个真实工程应用的 PLC 控制系统不能脱离低压电器而独立存在。低压电器通常是指工作在交流 50 Hz（60 Hz）、额定电压小于 1200 V 和直流额定电压小于 1500 V 的电路中，起

通断、保护、控制或调节作用的电器。

低压电器的分类方法很多，按照不同的分类方式有不同的类型。按照用途分类如下：

（1）控制电器

控制电器主要用于电力拖动和自动控制系统，包括继电器、接触器、主令电器、起动器、控制器和电磁铁等。

（2）配电电器

配电电器主要用于低压配电系统和动力装置中，包括刀开关、转换开关、断路器和熔断器等，要求在系统发生故障的情况下动作准确，工作可靠，有足够的热稳定性和动稳定性。

这部分内容是学习 PLC 的必备知识，如继电器、按钮和断路器等低压电器是几乎所有的 PLC 控制系统都要使用的，但由于内容较多，且相对比较简单，请读者应提前学习，因此在此不做赘述。以下将对一些读者不易掌握而且重要的器件分别重点介绍。

## 1.3.2　传感器和变送器

传感器在 PLC 控制系统中很常用，但在使用中有一定的难度。

传感器（transducer/sensor）是一种检测装置，能感受到被测量的信息，并能将感受到的信息按一定规律变换成为电信号或其他所需形式的信息输出，以满足信息的传输、处理、存储、显示、记录和控制等要求。

**1. 传感器的分类**

传感器的分类方法较多，常见的分类如下：

（1）按用途分类

按用途分类有压力传感器、位置传感器、液位传感器、能耗传感器、速度传感器、加速度传感器、射线辐射传感器以及热敏传感器。

（2）按原理分类

按原理分类有振动传感器、湿敏传感器、磁敏传感器、气敏传感器、真空度传感器以及生物传感器等。

（3）按输出信号分类

按输出信号分类有模拟式传感器、数字式传感器与开关式传感器。

模拟式传感器：将被测量的非电学量转换成模拟电信号。

数字式传感器：将被测量的非电学量转换成数字输出信号（包括直接和间接转换）。

开关式传感器：当一个被测量的信号达到某个特定的阈值时，传感器相应地输出一个设定的低电平或高电平信号。

**2. 开关式传感器的应用**

开关式传感器就是接近开关。接近式位置开关是与（机器的）运动部件无机械接触而能操作的位置开关。当运动的物体靠近开关到一定位置时，开关发出信号，达到行程控制及计数自动控制的开关。也就是说，它是一种非接触式无触头的位置开关，是一种开关型的传感器，简称接近开关（proximity sensors），又称接近传感器，外形如图 1-15 所示。接近开关有行程开关、微动开关的特性，又有传感性

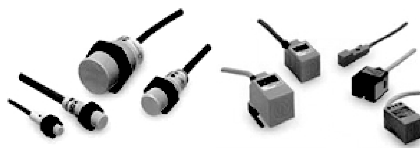


图 1-15　接近开关

能，而且动作可靠，性能稳定，频率响应快，使用寿命长，抗干扰能力强等。它由感应头、高频振荡器、放大器和外壳组成。常见的接近开关有 LJ、CJ 和 SJ 等系列产品。

（1）接近开关的功能

当运动部件与接近开关的感应头接近时，就使其输出一个电信号。接近开关在电路中的作用与行程开关相同，都是位置开关，起限位作用，但两者是有区别的：行程开关有触头，是接触式的位置开关；而接近开关是无触头的，是非接触式的位置开关。

（2）接近开关的分类和工作原理

按照工作原理区分，接近开关分为电感式、电容式、光电式和磁感式等形式。另外，根据应用电路电流的类型分为交流型和直流型。

① 电感式接近开关的感应头是一个具有铁氧体磁芯的电感线圈，只能用于检测金属体，在工业中应用非常广泛。振荡器在感应头表面产生一个交变磁场，当金属块接近感应头时，金属中产生的涡流吸收了振荡的能量，使振荡减弱以至停振，因而产生振荡和停振两种信号，经整形放大器转换成二进制的开关信号，从而起到"开""关"的控制作用。通常把接近开关刚好动作时感应头与检测物体之间的距离称为动作距离。

② 电容式接近开关的感应头是一个圆形平板电极，与振荡电路的地线形成一个分布电容，当有导体或其他介质接近感应头时，电容量增大而使振荡器停振，经整形放大器输出电信号。电容式接近开关既能检测金属，又能检测非金属及液体。电容式传感器体积较大，而且价格要贵一些。

③ 磁感式接近开关主要指霍尔接近开关，霍尔接近开关的工作原理是霍尔效应，当带磁性的物体靠近霍尔开关式时，霍尔接近开关的状态翻转（如由"ON"变为"OFF"）。有的资料上将干簧继电器也归类为磁性接近开关。

④ 光电式传感器是根据投光器发出的光在检测体上发生光量的增减，用光电变换元件组成的受光器检测物体有无或大小的非接触式控制器件。光电式传感器的种类很多，按照其输出信号的形式，可以分为模拟式、数字式和开关量输出式。

利用光电效应制成的传感器称为光电式传感器。光电式传感器的种类很多，其中，输出形式为开关量的传感器为光电式接近开关。

光电式接近开关主要由光发射器和光接收器组成。光发射器用于发射红外光或可见光，光接收器用于接收发射器发射的光，并将光信号转换成电信号，以开关量形式输出。

按照接收器接收光的方式不同，光电式接近开关可以分为对射式、反射式和漫射式 3 种。光发射器和光接收器有一体式和分体式两种形式。

⑤ 此外，还有特殊种类的接近开关，如光纤接近开关和气动接近开关。特别是光纤接近开关在工业上使用越来越多，它非常适合在狭小的空间、恶劣的工作环境（高温、潮湿和干扰大）、易爆环境以及准确度要求高等条件下使用。光纤接近开关的问题是价格相对较高。

（3）接近开关的选型

常用的电感式接近开关（inductive sensor）型号有 LJ 系列产品，电容式接近开关（capacitive sensor）型号有 CJ 系列产品，磁感式接近开关有 HJ 系列产品，光电型接近开关有 OJ 系列。当然，还有很多厂家都有自己的产品系列，一般接近开关型号的含义如图 1-16 所示。接近开关的选择要遵循以下原则：

① 接近开关类型的选择。检测金属时优先选用感应式接近开关，检测非金属时选用电

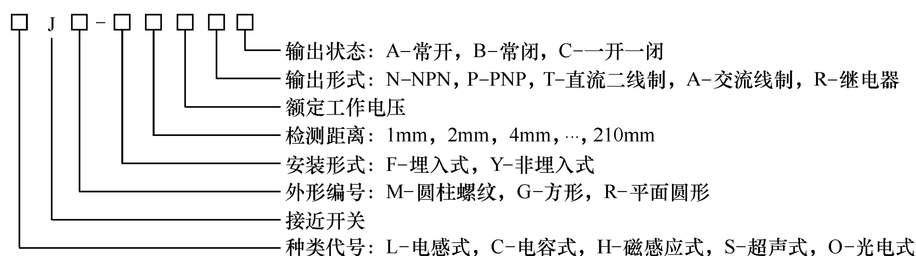



图 1-16　接近开关型号的含义

容式接近开关，检测磁信号时选用磁感式接近开关。

② 外观的选择。根据实际情况选用，但圆柱螺纹形状的最为常见。

③ 检测距离（sensing range）的选择。根据需要选用，但注意同一接近开关检测距离并非恒定，接近开关的检测距离与被检测物体的材料、尺寸及物体的移动方向有关。表 1-3 列出了目标物体材料对于检测距离的影响。不难发现，感应式接近开关对于有色金属的检测明显不如检测钢和铸铁。常用的金属材料不影响电容式接近开关的检测距离。

**表 1-3　目标物体材料对检测距离的影响**

| 序　　号 | 目标物体材料 | 影响系数 | |
|---|---|---|---|
| | | 感　应　式 | 电　容　式 |
| 1 | 碳素钢 | 1 | 1 |
| 2 | 铸铁 | 1.1 | 1 |
| 3 | 铝箔 | 0.9 | 1 |
| 4 | 不锈钢 | 0.7 | 1 |
| 5 | 黄铜 | 0.4 | 1 |
| 6 | 铝 | 0.35 | 1 |
| 7 | 紫铜 | 0.3 | 1 |
| 8 | 水 | 0 | 0.9 |
| 9 | PVC（聚氯乙烯） | 0 | 0.5 |
| 10 | 玻璃 | 0 | 0.5 |

目标的尺寸同样对检测距离有影响。满足以下一个条件时，检测距离不受影响。

- 当检测距离的 3 倍大于接近开关感应头的直径，而且目标物体的尺寸大于或等于 3 倍的检测距离 ×3 倍的检测距离（长 × 宽）。
- 当检测距离的 3 倍小于接近开关感应头的直径，而且目标物体的尺寸大于或等于检测距离 × 检测距离（长 × 宽）。

如果目标物体的面积达不到推荐数值时，接近开关的有效检测距离将按照表 1-4 推荐的数值减少。

**表 1-4　目标物体的面积对检测距离的影响**

| 占推荐目标面积的比例 | 影响系数 | 占推荐目标面积的比例 | 影响系数 |
|---|---|---|---|
| 75% | 0.95 | 25% | 0.85 |
| 50% | 0.90 | | |

④ 信号的输出选择。交流接近开关输出交流信号，而直流接近开关输出直流信号。注意，负载的电流一定要小于接近开关的输出电流，否则应添加转换电路解决。接近开关的信号输出能力见表 1–5。

**表 1–5　接近开关的信号输出能力**

| 接近开关种类 | 输出电流/mA | 接近开关种类 | 输出电流/mA |
|---|---|---|---|
| 直流二线制 | 50 ~ 100 | 直流三线制 | 150 ~ 200 |
| 交流二线制 | 200 ~ 350 | | |

⑤ 触头数量的选择。接近开关有常开触头和常闭触头。可根据具体情况选用。

⑥ 开关频率的确定。开关频率是指接近开关每秒从“开”到“关”转换的次数。直流接近开关可达 200 Hz；而交流接近开关要小一些，只能达到 25 Hz。

⑦ 额定电压的选择。对于交流型的接近开关，优先选用 AC 220 V 和 AC 36 V，而对于直流型的接近开关，优先选用 DC 12 V 和 DC 24 V。

（4）应用接近开关的注意事项

① 单个 NPN 型和 PNP 型接近开关的接线。在直流电路中使用的接近开关有二线式（两根导线）、三线式（3 根导线）和四线式（4 根导线）等多种，二线、三线和四线式接近开关都有 NPN 型和 PNP 型两种，通常日本和美国多使用 NPN 型接近开关，欧洲多使用 PNP 型接近开关，而我国则二者都有应用。NPN 型和 PNP 型接近开关的接线方法不同，正确使用接近开关的关键就是正确接线，这一点至关重要。

接近开关的导线有多种颜色，通常 BN 表示棕色的导线；BU 表示蓝色的导线；BK 表示黑色的导线；WH 表示白色的导线；GR 表示灰色的导线。根据国家标准，各颜色导线的作用按照表 1–6 定义。对于二线式 NPN 型接近开关，棕色线与负载相连，蓝色线与零电位点相连；对于二线式 PNP 型接近开关，棕色线与高电位相连，负载的一端与接近开关的蓝色线相连，而负载的另一端与零电位点相连。如图 1–17 和图 1–18 所示分别为二线式 NPN 型接近开关接线图和二线式 PNP 型接近开关接线图。

**表 1–6　接近开关的导线颜色定义**

| 种　类 | 功　能 | 接线颜色 | 端　子　号 |
|---|---|---|---|
| 交流二线式和直流二线式（不分极性） | NO（接通） | 不分正负极，颜色任选，但不能为黄色、绿色或者黄绿双色 | 3、4 |
| | NC（分断） | | 1、2 |
| 直流二线式（分极性） | NO（接通） | 正极棕色，负极蓝色 | 1、4 |
| | NC（分断） | 正极棕色，负极蓝色 | 1、2 |
| 直流三线式（分极性） | NO（接通） | 正极棕色，负极蓝色，输出黑色 | 1、3、4 |
| | NC（分断） | 正极棕色，负极蓝色，输出黑色 | 1、3、2 |
| 直流四线式（分极性） | 正极 | 棕色 | 1 |
| | 负极 | 蓝色 | 3 |
| | NO 输出 | 黑色 | 4 |
| | NC 输出 | 白色 | 2 |

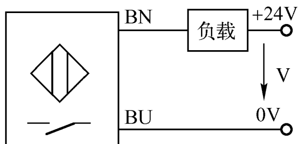



图 1-17　二线式 NPN 型接近开关接线图

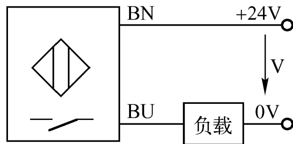



图 1-18　二线式 PNP 型接近开关接线图

表 1-6 中的“NO”表示常开、输出，而“NC”表示常闭、输出。

对于三线式 NPN 型接近开关，棕色的导线与一端负载，同时与电源正极相连；黑色的导线是信号线，与负载的另一端相连；蓝色的导线与电源负极相连。对于三线式 PNP 型接近开关，棕色的导线与电源正极相连；黑色的导线是信号线，与负载的一端相连；蓝色的导线与负载的另一端及电源负极相连，如图 1-19 和图 1-20 所示。

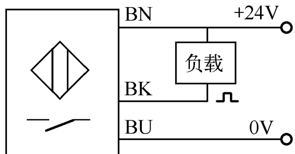



图 1-19　三线式 NPN 型接近开关接线图

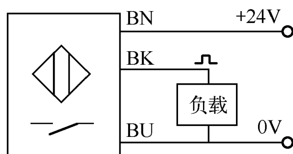



图 1-20　三线式 PNP 型接近开关接线图

四线式接近开关的接线方法与三线式接近开关类似，只不过，四线式接近开关多了一对触头而已，其接线图如图 1-21 和图 1-22 所示。

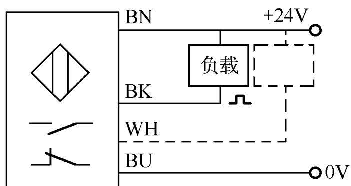



图 1-21　四线式 NPN 型接近开关接线图

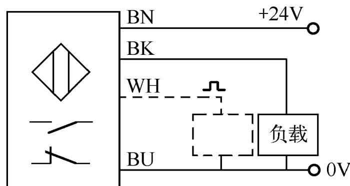



图 1-22　四线式 PNP 型接近开关接线图

② 单个 NPN 型和 PNP 型接近开关的接线常识。初学者经常不能正确区分 NPN 型和 PNP 型的接近开关，其实只要记住一点：PNP 型接近开关是正极开关，也就是信号从接近开关流向负载；而 NPN 型接近开关是负极开关，也就是信号从负载流向接近开关。

**【例 1-2】**在图 1-23 中，有一只 NPN 型接近开关与指示灯相连，当一个铁块靠近接近开关时，回路中的电流会怎样变化？

**解：**指示灯就是负载，当铁块到达接近开关的感应区时，回路突然接通，指示灯由暗变亮，电流从很小变化到 100% 的幅度，电流曲线如图 1-24 所示（理想状况）。




图 1-23　接近开关与指示灯相连的示意图

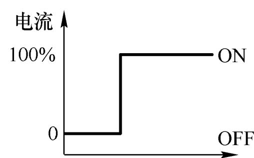



图 1-24　回路电流变化曲线

【**例 1-3**】某设备用于检测 PVC 物块，当检测物块时，设备上的 DC 24 V 功率为 12 W 的报警灯亮，请选用合适的接近开关，并画出原理图。

**解**：因为检测物体的材料是 PVC，所以不能选用感应接近开关，但可选用电容式接近开关。报警灯的额定电流为 $I_N=\frac{P}{U}=\frac{12}{24}=0.5\ A$，查表 1-6 可知，直流接近开关承受的最大电流为 0.2 A，所以采用图 1-20 的方案不可行，信号必须进行转换，原理图如图 1-25 所示，当物块靠近接近开关时，黑色的信号线上产生高电平，其负载继电器 KA 的常开得电，继电器 KA 的常开触头闭合，所以报警灯 EL 亮。

由于没有特殊规定，所以 PNP 或 NPN 型接近开关以及二线或三线式接近开关都可以选用。本例选用三线式 PNP 型接近开关。

**3. 变送器简介**

变送器（transmitter）是把传感器的输出信号转变为可被控制器识别的信号（或将传感器输入的非电量转换成电信号同时放大以便供远方测量和控制的信号源）的转换器。传感器和变送器一同构成自动控制的监测信号源。不同的物理量需要不同的传感器和相应的变送器。变送器的种类很多，用在工控仪表上的变送器主要有温度变送器、压力变送器、流量变送器、电流变送器和电压变送器等。变送器常与传感器做成一体，也可独立于传感器单独作为商品出售，如压力变送器和温度变送器等。一种变送器如图 1-26 所示。

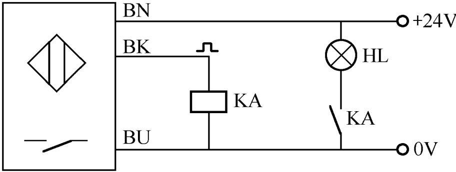
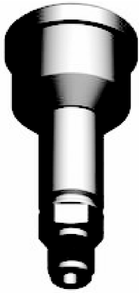


图 1-25　原理图


图 1-26　变送器

**4. 传感器和变送器应用**

变送器按照接线分有三种：两线式、三线式、四线式。

两线式的变送器两根线既是电源线又是信号线；三线式的变送器两根线是信号线（其中一根共 GND），一根线是电源正线；四线式的两根线是电源线，两根线是信号线（其中一根共 GND）。

两线式的变送器具有不易受寄生热电偶和沿电线电阻压降和温漂的影响、可用非常便宜的更细的导线，可节省大量电缆线和安装费用等优点，不具备上述优点的三线制和四线制变送器将被两线制变送器所取代。

（1）S7 -300 CPU 模块与四线式电压变送器接法

四线式电压变送器接法相对容易，两根线为电源线，两根线为信号线，接线图如图 1-27所示。

（2）S7 -300 CPU 模块与四线式电流变送器接法

四线式电流变送器接法相对容易，两根线为电源线，两根线为信号线，接线图如

图 1-28所示。

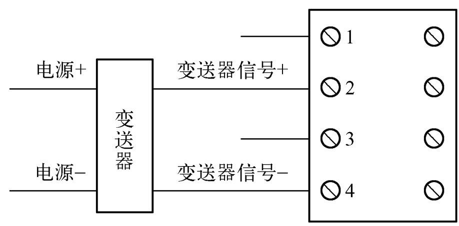



图 1-27　四线式电压变送器接线

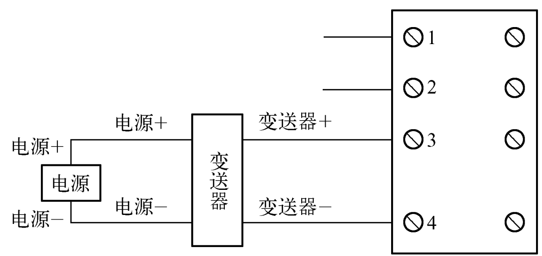



图 1-28　四线式电流变送器接线

（3）S7 - 300 CPU 模块与三线式电流变送器接法

三线式电压变送器，两根线为电源线，一根线为信号线，其中信号负（变送器负）和电源负为同一根线，接线图如图 1-29 所示。

（4）S7 - 300 CPU 模块与二线式电流变送器接法

二线式电流变送器接法容易接错，两根线为电源线，同时也为信号线，接线图如图 1-30 所示。

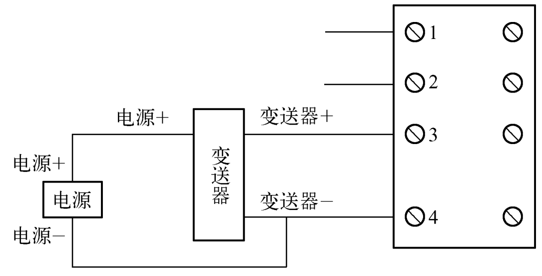



图 1-29　三线式电流变送器接线

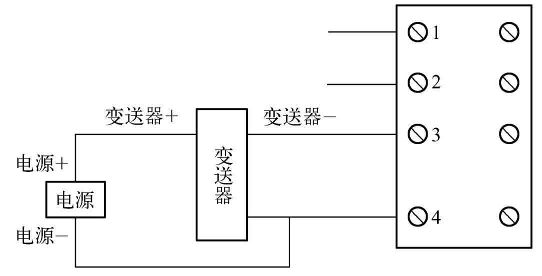



图 1-30 二线式电流变送器接线

### 1. 3. 3　隔离器

隔离器是一种采用线性光耦隔离原理，将输入信号进行转换输出的器件。输入、输出和工作电源三者相互隔离，特别适合与需要电隔离的设备、仪表配合使用。隔离器又名信号隔离器，是工业控制系统中的重要组成部分。某品牌的隔离器如图 1-31 所示。

在 PLC 控制系统中，隔离器最常用于传感器与 PLC 的模拟量输入模块之间，以及执行器与 PLC 的模拟量输出模块之间，起抗干扰和保护模拟量模块的作用。隔离器的一个应用实例如图 1-32 所示。



图 1-31　隔离器外形

### 1. 3. 4　浪涌保护器

浪涌保护器（电涌保护器）又称防雷器，简称（SPD），适用于交流 50/60 Hz、额定电

压 220 V ~ 380 V 的供电系统（或通信系统）中，对间接雷电和直接雷电影响或其他瞬时过压的电涌进行保护。它是一种保护电器。其外形如图 1-33 所示。

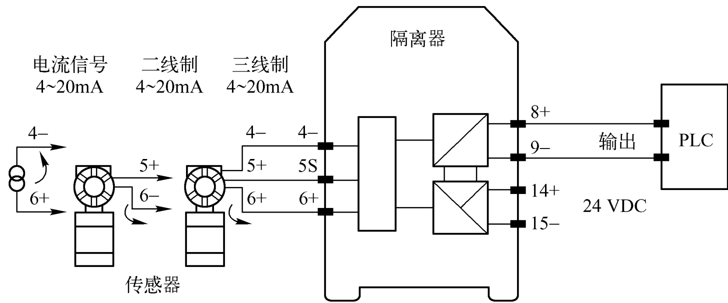



图 1-32　隔离器应用实例




图 1-33　浪涌保护器外形

浪涌保护器有信号浪涌保护器、直流电源浪涌保护器和交流电源浪涌保护器，主要用于防雷。

浪涌保护器的一个应用实例如图 1-34 所示。

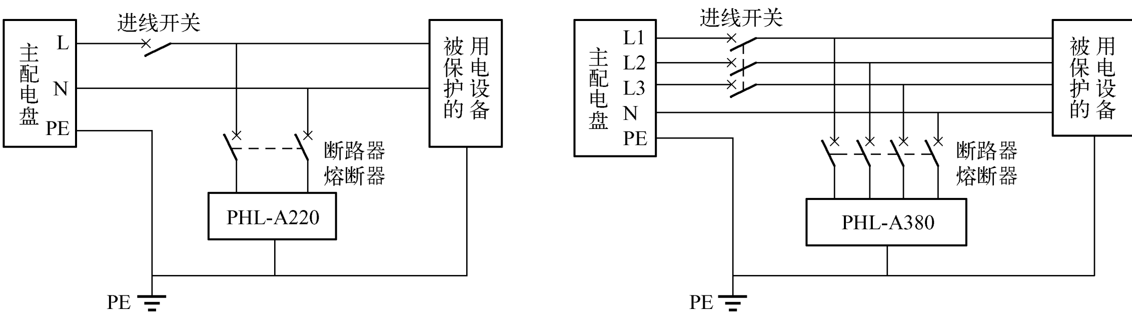



图 1-34　浪涌保护器应用实例

## 1.3.5　安全栅

安全栅（safety barrier），接在本质安全电路和非本质安全电路之间。将供给本质安全电路的电压和电流限制在一定安全范围内的装置。安全栅又称安全限能器。

本安型安全栅应用在本安防爆系统的设计中，它是安装于安全场所并含有本安电路和非本安电路的装置，电路中通过限流和限压电路，限制了送往现场本安回路的能量，从而防止非本安电路的危险能量串入本安电路，它在本安防爆系统中称为关联设备，是本安系统的重要组成部分。安全栅的外形如图 1-35 所示。



图 1-35　安全栅外形

安全栅的一个应用实例如图 1-36 所示。

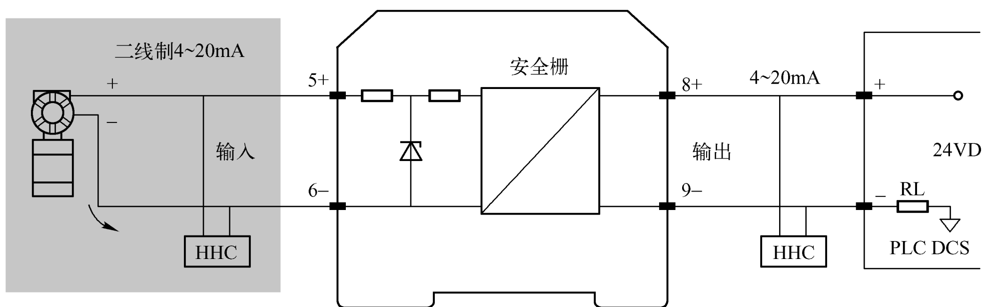



图 1−36　安全栅应用实例

# 第 2 章　S7－300/400 PLC 的硬件介绍

本章介绍常用 S7－300/400 的 CPU 模块、数字量输入/输出模块、模拟量输入/输出模块、功能模块、机架和电源模块的功能、接线和安装，这些内容是后续程序设计和控制系统设计所必须掌握的前导知识。

## 2.1　S7－300 PLC 概述

### 2.1.1　西门子 PLC 简介

德国西门子（SIEMENS）公司是欧洲最大的电子和电气设备制造商之一，其生产的 SIMATIC（“SIEMENS Automatic”即西门子自动化）可编程序控制器在欧洲处于领先地位。

西门子公司的第一代 PLC 是 1975 年投放市场的 SIMATIC S3 系列的控制系统。之后在 1979 年，西门子公司将微处理器技术应用到 PLC 中，研制出了 SIMATIC S5 系列，取代了 S3 系列，目前 S5 系列产品仍然有小部分在工业现场使用，20 世纪末，又在 S5 系列的基础上推出了 S7 系列产品。

SIMATIC S7 系列产品分为 S7－200、S7－200CN、S7－200 SMART、S7－1200、S7－300、S7－400 和 S7－1500 共七个产品系列。S7－200 是在西门子收购的小型 PLC 的基础上发展而来，因此其指令系统、程序结构和编程软件和 S7－300/400 有较大的区别，在西门子 PLC 产品系列中是一个特殊的产品。S7－200 SMART 是 S7－200 的升级版本，是西门子家族的新成员，于 2012 年 7 月发布，其绝大多数的指令和使用方法与 S7－200 类似，其编程软件也和 S7－200 的类似，而且在 S7－200 运行的程序，大部分可以在 S7－200 SMART 中运行。S7－1200 系列是在 2009 年才推出的新型小型 PLC，定位于 S7－200 和 S7－300 产品之间。S7－300/400 是由西门子的 S5 系列发展而来，是西门子公司的最具竞争力的 PLC 产品。2012 年西门子公司又推出了新品 S7－1500 系列产品。西门子的 PLC 产品系列的定位见表 2-1。

表 2-1　SIMATIC S7 控制器的定位

| 序　号 | 控 制 器 | 定　位 | 主要任务和性能特征 |
|---|---|---|---|
| 1 | S7－200 和 S7－200CN | 低端的离散自动化系统和独立自动化系统中使用的紧凑型辑控制器模块 | 串行模块结构、模块化扩展<br>紧凑设计，CPU 集成 I/O<br>实时处理能力，高速计数器和报警输入和中断<br>易学易用的软件<br>多种通信选项 |
| 2 | S7－200 SMART | 低端的离散自动化系统和独立自动化系统中使用的紧凑型辑控制器模块，是 S7－200 的升级版本 | 串行模块结构、模块化扩展<br>紧凑设计，CPU 集成 I/O<br>集成了 PROFINET 接口<br>实时处理能力，高速计数器和报警输入和中断<br>易学易用的软件<br>多种通信选项 |

（续）

| 序　号 | 控 制 器 | 定　位 | 主要任务和性能特征 |
|---|---|---|---|
| 3 | S7－1200 | 低端的离散自动化系统和独立自动化系统中使用的小型控制器模块 | 可升级及灵活的设计<br>集成了 PROFINET 接口<br>集成了强大的计数、测量、闭环控制及运动控制功能<br>直观高效的 STEP 7 Basic 工程系统可以直接组态控制器和 HMI |
| 4 | S7－300 | 中端的离散自动化系统中使用的控制器模块 | 通用型应用和丰富的 CPU 模块种类<br>高性能<br>模块化设计，紧凑设计<br>由于使用 MMC 存储程序和数据，系统免维护 |
| 5 | S7－400 | 高端的离散和过程自动化系统中使用的控制器模块 | 特别高的通信和处理能力<br>定点加法或乘法的指令执行速度最快为 0.03 μs<br>大型 I/O 框架和最高 20 MB 的主内存<br>快速响应，实时性强，垂直集成<br>支持热插拔和在线 I/O 配置，避免重启<br>具备等时模式，可以通过 PROFIBUS 控制高速机器 |
| 6 | S7－1500 | 中高端系统 | S7－1500 控制器除了包含多种创新技术之外，还设定了新标准，最大程度提高生产效率。无论是小型设备还是对速度和准确性要求较高的复杂设备装置，都一一适用。SIMATIC S7－1500 无缝集成到 TIA 博途中，极大提高了工程组态的效率 |

SIMATIC 产品除了 SIMATIC S7 外，还有 M7、C7 和 WinAC 系列。

SIMATIC C7 是基于 S7－300 系列 PLC 性能，同时集成了 HMI，具有节省空间的特点。

SIMATIC M7－300/400 采用了与 S7－300/400 相同的结构，又具有兼容计算机的功能，可以用 C、C++等高级语言编程，SIMATIC M7－300/400 适于需要大数据量处理和实时性要求高的场合。

WinAC 是在个人计算机上实现 PLC 功能，突破了传统 PLC 开放性差、硬件昂贵等缺点，WinAC 具有良好的开放性和灵活性，可以很方便集成第三方的软件和硬件。

### 2.1.2　S7－300 的基本结构

S7－300 系列 PLC 是模块化结构设计的 PLC，各个单独模块之间可进行广泛组合和扩展。它的主要组成部分有电源模块（PS）、中央处理器模块（CPU）、导轨（RACK）、接口模块（IM）、信号模块（SM）和功能模块（FM）等。S7－300 可以通过 MPI 接口直接与编程器（PG）、操作面板（OP）和其他 S7 系列 PLC 相连。其实物如图 2-1 所示，系统构成如图 2-2 所示。

① 电源模块（PS）。电源模块用于向 CPU 及其扩展模块提供 DC ＋24 V 电源。电源模块与 CPU 之间用电缆（连接器）连接，而不是用背板供电，因此有的设计者为了省钱用普通开关电源取代西门子的电源模块，但这样做是不被西门子公司推荐的。目前电源模块有 2 A、5 A 和 10 A 等规格，可根据实际选用。

② 中央处理器模块（CPU）。S7－300 的 CPU 模块主要包括 CPU312、CPU312C、CPU313C、CPU313C－PtP 及 CPU314－2DP 等型号，有的型号还有不同的版本号（如 CPU314－2DP 目前有 2.0 版和 2.6 版），每种 CPU 有其不同的性能。CPU 型号中的“C”表

示紧凑型 CPU；CPU 型号中的“DP”表示带有 9 针的 DP 接口；CPU 型号中的“PtP”表示带有 15 针的 PtP 接口；有的 CPU 上还带有输入/输出端子。

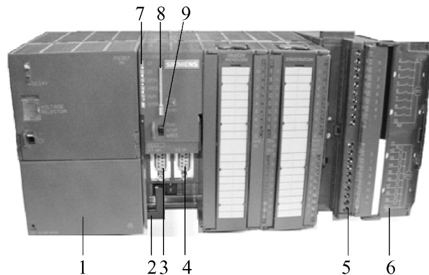



图 2-1　S7-300 PLC 实物图

1—电源模块　2—DC 24 V 连接器　3—MPI 接口
4—DP 接口　5—前连接器　6—前盖　7—状态和故障指示灯　8—存储卡　9—模式开关

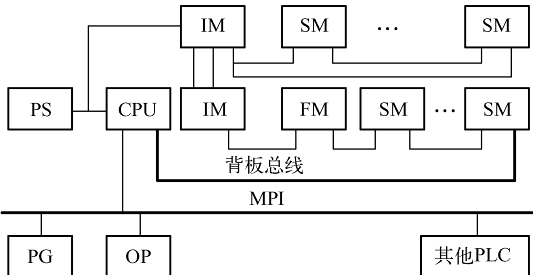



图 2-2　S7-300 PLC 系统构成图

③ 导轨（RACK）。导轨是安装 S7-300 PLC 各类模块的机架，它是特制的异形板，其标准长度有 160 mm、482 mm、530 mm、830 mm 和 2000 mm，可以根据实际选用。此导轨可以切割成需要的长度。

④ 信号模块（SM）。信号模块是数字量 I/O 模块和模拟量 I/O 模块的总称。信号模块主要有 SM321（数字量输入）、SM322（数字量输出）、SM331（模拟量输入）和 SM332（模拟量输出）等模块。每个模块都带有一个总线连接器，用于 CPU 和其他模块之间的数据通信。

⑤ 功能模块（FM）。功能模块主要用于对实时性和存储量要求高的控制任务。如计数模块 FM350、定位模块 FM353 等。

⑥ 通信处理模块（CP）。通信处理模块用于 PLC 之间、PLC 与计算机和其他智能设备之间的通信，可以将 PLC 接入工业以太网、PROFIBUS 和 AS-I 网络或用于串行通信。它可以减轻 CPU 处理通信的负担，并减少用户对通信功能的编程工作。

⑦ 接口模块（IM）。接口模块用于多机架配置时连接主机架（CR）和扩展机架（ER）。S7-300 PLC 通过分布式的主机架和连接的扩展机架（最多可连接三个扩展机架），可以操作最多 32 个模块。

## 2.2　S7-300 的 CPU 模块及其接线

S7-300 的 CPU 模块共有几十个不同的型号，按照性能等级划分，可涵盖各种应用领域。以下将介绍 CPU 的技术规范和接线

### 2.2.1　CPU 模块的技术规范

**1. CPU 模块的分类**

① 紧凑型 CPU，包括 CPU 312C、313C、313C-PtP、313C-2DP、314C-PtP 和 314C-2DP。各 CPU 均有计数、频率测量和脉冲宽度调制功能。有的具备定位功能，有的带有

I/O。紧凑型 CPU 外形如图 2-3 所示。

② 标准型 CPU，包括 CPU 312、313、314、315、315－2DP 和 316－2DP 等。标准型 CPU 外形如图 2-4 所示。标准型 CPU 不带 I/O，所以应用时需要增加信号模块。

③ 户外型 CPU，包括 CPU 312 IFM、314 IFM、314 户外型和 315－2DP。在恶劣的环境下使用。

④ 高端 CPU，包括 317－2DP 和 CPU 318－2DP。具有大容量的程序存储器和 PROFIBUS－DP 接口，可以用于大规模的 I/O 配置（如 CPU 318－2DP 最多可以配置 65536 个数字 I/O），建立分布式 I/O 结构。

⑤ 故障安全型 CPU，如 CPU 315F，不需要对故障 I/O 进行额外接线，可以组态成一个故障安全型自动化系统。

⑥ 特殊性 CPU，如 CPU 317T－2DP，集成了大量控制功能，有的书上称 CPU 317T－2DP 技术型 CPU，其实际具有运动控制功能。CPU 317－2 PN/DP（有 PN 接口）是高端的 CPU，其性能优越，价格也比较高，带 PN 口的模块将越来越多。工艺型 CPU 外形如图 2-5 所示。



图 2-3　紧凑型 CPU 外形



图 2-4　标准型 CPU 外形

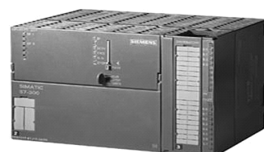


图 2-5　工艺型 CPU 外形

**2. CPU 的状态与故障显示 LED**

CPU 317－2DP 的面板如图 2-6 所示，其他 CPU 的面板和 CPU 317－2DP 类似。

SF（系统出错/故障显示，红色）：CPU 硬件故障或软件错误时亮。

BATF（电池故障，红色）：电池电压低或没有电池时亮。

DC 5 V（+5 V 电源指示，绿色）：5 V 电源正常时亮。

FRCE（强制，黄色）：至少有一个 I/O 被强制时亮。

RUN（运行方式，绿色）：CPU 处于 RUN 状态时亮；重新起动时以 2 Hz 的频率闪亮；HOLD（单步、断点）状态时以 0.5 Hz 的频率闪亮。

STOP（停止方式，黄色）：CPU 处于 STOP 状态，HOLD 状态或重新起动时常亮。

BF（总线错误，红色）：总线出错时亮。

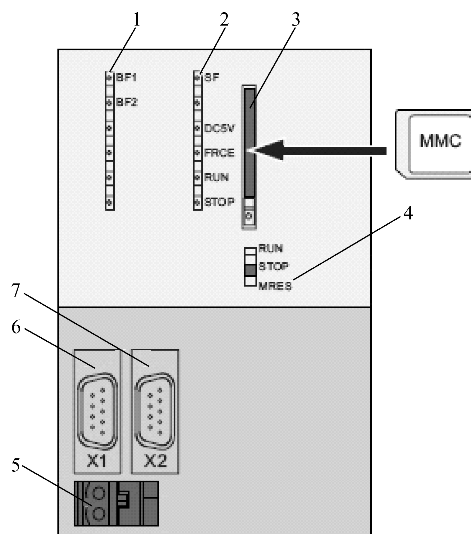



图 2-6　CPU 317－2DP 的面板
1—总线故障指示器　2—状态和错误显示　3—微存储卡（MMC）的插槽　4—模式选择器开关　5—电源连接　6—接口 X1（MPI/DP）　7—接口 X2（DP）

**3. 模式选择开关**

模式选择开关的外形如图 2-7 所示。




图 2-7　模式选择开关的外形

① RUN 模式，CPU 执行用户程序。

② STOP 模式，CPU 不执行用户程序。

③ MRES，CPU 存储器复位，带有用于 CPU 存储器复位的按钮功能的模式选择器开关位置。

通过模式选择器开关进行 CPU 存储器复位需要特定操作顺序：通电后从 STOP 位置扳到 MRES 位置，“STOP” LED 熄灭 1 s，亮 1 s，再熄灭 1 s 后保持亮。放开开关，使它回到 STOP 位置，然后又回到 MRES，“STOP” LED 以 2 Hz 的频率至少闪动 3 s，表示正在执行复位，最后“STOP” LED 一直亮。

**4. MMC 微存储卡**

基于 FEPROM 的微存储卡称为 MMC，其外形如图 2-8 所示，MMC 不是 CPU 的标准配置，需要另外订货。MMC 卡用于断电时保存用户数据。MMC 用于 S7、C7 和 ET200S 的 CPU 的装载存储器，程序和数据下载后保存在 MMC 中。如果没有插入 MMC 卡是不能下载程序的。不能带电插拔 MMC，否则会损坏 MMC 卡。西门子 PLC 必须使用专用的 MMC 卡，不可以普通数码产品的存储卡代替。




图 2-8　MMC 存储卡外形

不能用 CPU 选择开关的操作来删除下载到 MMC 的系统数据和程序。为了删除存储卡中的程序，首先必须建立 PLC 和计算机之间的连接，在 SIMATIC 管理器的工具栏上，单击在线按钮，打开在线视图，选中要删除的块，按键盘上的“Delete”（删除键），即可删除块文件夹中的数据。但注意 CPU 中集成的 SFC 和 SFB 不能被删除。

CPU 的存储区由装载存储器、工作存储器和系统存储器组成。工作存储器类似于计算机的内存条，装载存储器类似于计算机的硬盘。以下分别介绍三种存储器：

① 装载存储器。装载存储器用于保存逻辑块（不包含符号地址和注释）、数据块和系统数据。下载程序时，用户程序下载到装载存储器，但符号表、注释不能下载，仍然保存在编程设备中。在 PLC 上电时，CPU 把装载存储器中的可执行的部分复制到工作存储器。而 PLC 断电时，需要保存的数据自动保存在装载存储器中。

② 工作存储器。工作存储器集成在 CPU 中的高速存取的 RAM 存储器，用于存储 CPU 运行时的用户程序和数据，如组织块、功能块等。用模式选择开关复位 CPU 的存储器时，RAM 中程序被清除，但 FEPROM 中的程序不会被清除。

③ 系统存储器。系统存储器是 CPU 为用户提供的存储组件，用于存储用户程序的操作数据，例如过程映像输入、过程映像输出、位存储、定时器、计数器、块堆栈和诊断缓冲区等。

**【例 2-1】** 某设备的控制器为 CPU315 -2PN/DP，问怎样选择 MMC 存储卡的容量。

**解：** MMC 存储卡的选择主要依据程序的容量，如果程序已经编写完毕，只要在 SIMATIC 管理器中选中程序的“块”，单击鼠标右键，在弹出的快捷菜单中选中并打开“对象属

性”，如图2-9所示，可以看到装载存储器的需要大小为28042 B，约为28 KB。

再查看表2-2，CPU315－2PN/DP的工作存储器RAM为256 KB，通常MMC的容量不应小于工作存储器的容量，因此选择MMC为256 KB。

**5. S7－300 CPU的技术参数**

S7－300 CPU的技术参数是设计选型的重要依据，因此读者在入门学习时，要养成查看技术参数的习惯，以下仅列出最常用的几个型号CPU的技术参数，见表2-2。




图2-9　对象属性

**表2-2　S7－300 CPU的技术参数表**

| CPU | 312 | 314 | 315－2DP | 315－2PN/DP | 317－2DP | 319－3PN/DP |
|---|---|---|---|---|---|---|
| 集成工作存储器RAM/KB | 64 | 96 | 128 | 256 | 512 | 1400 |
| 装载存储器（MMC）/MB | 最大4 | 最大8 | 最大8 | 最大8 | 最大8 | 最大8 |
| 位操作执行时间/μs<br>浮点运算指令执行时间/μs | 0.2<br>3 | 0.2<br>3 | 0.1<br>2 | 0.1<br>2 | 0.05<br>1 | 0.01<br>0.04 |
| 位存储器（M）/B | 256 | 256 | 2048 | 2048 | 4096 | 8192 |
| S7定时器/计数器 | 128/128 | 256/256 | 256/256 | 256/256 | 512/512 | 2048/2048 |
| 最大数字量I/O点数<br>最大模拟量I/O点数 | 256/256<br>64/64 | 1024/1024<br>256/256 | 16384/16384<br>1024/1024 | 16384/16384<br>1024/1024 | 6553565535<br>4096/4096 | 6553565535<br>4096/4096 |
| 最大机架数/模块数<br>内置/经CP的DP接口数 | 1/8<br>0/4 | 4/32<br>0/4 | 4/32<br>1/4 | 4/32<br>1/4 | 4/32<br>2/4 | 4/32<br>1/4 |
| FB的最大块数/最大容量<br>FC的最大块数/最大容量<br>DB的最大块数/最大容量<br>OB的最大容量 | 1024/32 KB<br>1024/32 KB<br>1024/32 KB<br>32 KB | 2048/64 KB<br>2048/64 KB<br>1024/64 KB<br>64 KB | 2048/64 KB<br>1024/64 KB<br>1024/64 KB<br>64 KB | 2048/64 KB<br>2048/64 KB<br>1024/64 KB<br>64 KB | 2048/64 KB<br>2048/64 KB<br>2048/64 KB<br>64 KB | 2048/64 KB<br>2048/64 KB<br>4096/64 KB<br>64 KB |

## 2.2.2　紧凑型CPU的接线

紧凑型CPU的接线基本类似，下面以CPU314C－2DP为例讲解紧凑型CPU的接线。

**1. 电源模块和CPU的接线**

电源模块和CPU接线比较简单，S7－300系列的接线都相同。先打开PS 307电源模块、CPU前面板和PS 307上的电缆夹。再将电源电缆连接到L1、N和PS 307的保护接地（PE）端。最后将PS 307上的较低端子M和L+连接到CPU上的M和L+端子。电源模块和CPU接线如图2-10所示。

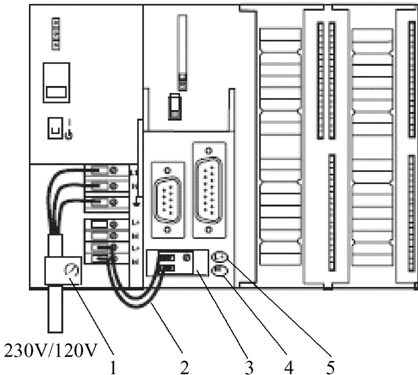



图2-10　电源模块和CPU接线

1—电缆夹　2—连接电缆　3—可拆卸的电源连接器　4—标记“M”　5—标记“L+”

**2. 数字I/O的接线**

紧凑型CPU一般带有数字I/O，输入为PNP型输入（高电平有效），这一点不同于S7－200系列

PLC，后者为 PNP 型和 NPN 型输入可选，因此输入端若要连接接近开关，通常应选择 PNP 型。输入端的 1 号端子与 DC 24 V 相连，而 20 号端子与 0 V 相连，10 号和 11 号端子不使用，其余输入端的端子都可以作为输入点使用。

输出也是 PNP 输出（高电平有效），这一点与 S7－200 系列 PLC 相同。输出端的 21 号和 31 号端子与 DC 24 V 相连，而 30 号和 40 号端子与 0 V 相连，其余输出端的端子都可以作为输出点使用。数字 I/O 的接线如图 2-11 所示。S7－300 PLC 的接线图一般印刷在机壳上，也可从西门子公司的网站上下载。

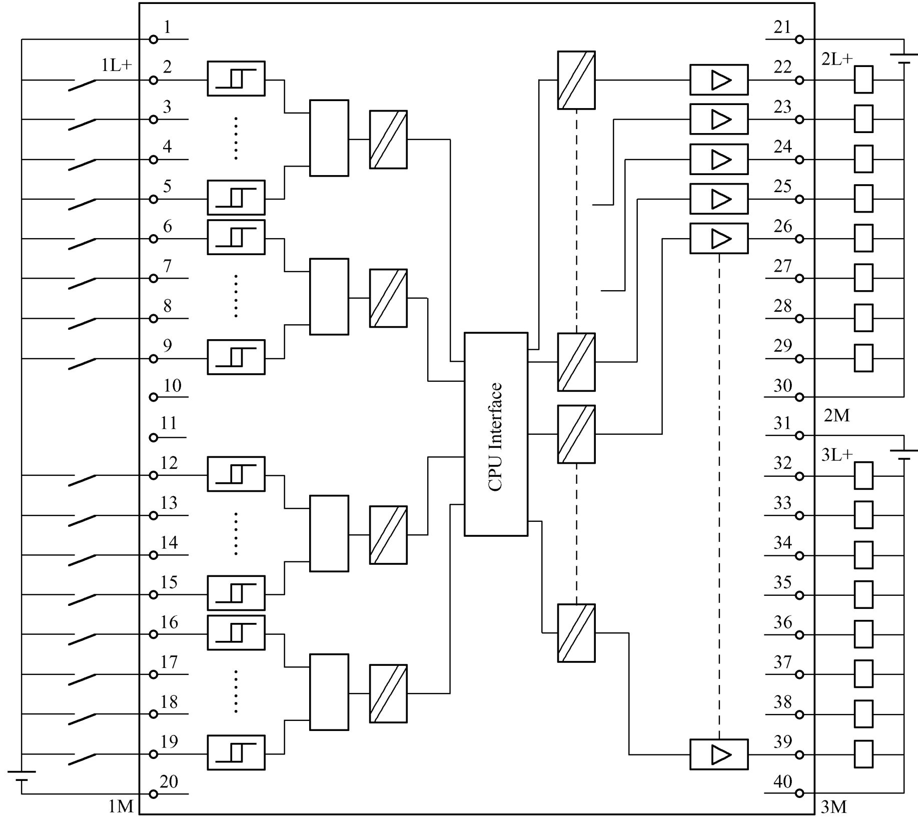



图 2-11　CPU314C－2DP 的数字 I/O 的接线

【例 2-2】某设备的控制器为 CPU314C－2DP，控制三相交流电动机的起停控制，并有一个接近开关限位，请设计接线图。

解：根据题意，只需要 3 个输入点和一个输出点，因此使用 CPU314C－2DP 上集成的 I/O 即可，输入端和输出端都是 PNP 型，因此接近开关只能用 PNP 型的接近开关（不用转换电路时），接线图如图 2-12 所示。交流电动机的起停一般要用交流接触器，交流回路由读者自行设计，在此不作赘述。

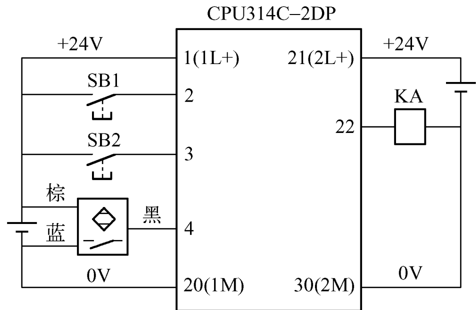



图 2-12　接线图

【例 2-3】某初学者按如图 2-13 所示接线，

并输入如图 2-14 所示的程序，压下按钮 SB1 后，发现，CPU 上指定的 Q0.0 指示灯不亮，监控程序，Q0.0 是导通的，分析原因。

**解**：CPU 上指定的 Q0.0 指示灯要接上电源（把 21 和 30 上接上电源）才会亮，这点与其他 PLC 不同（S7－200 不接负载电源，指示灯能亮）。

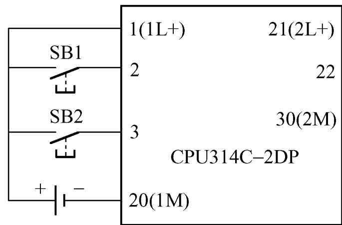


图 2-13　接线图

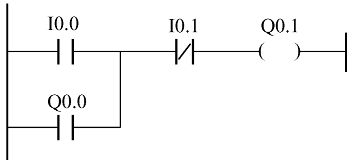


图 2-14　程序

**3. 模拟量 I/O 的接线**

紧凑型 CPU 除了带有数字 I/O，一般还自带模拟 I/O，这无疑是非常方便的，对于要求不高的控制系统，使用紧凑型 CPU 是非常经济的。

如图 2-15 所示，CPU314C－2DP 有 5 个模拟输入通道，其中的输入通道 0（CH0）、输入通道 1（CH1）、输入通道 2（CH2）和输入通道 3（CH3），都可以输入电压或者电流信号，但二者只能选择其一，不能同时输入。例如，如果要在通道 0 中采集电压信号，那么只要将电压信号连接在 2 号端子和 4 号端子之间即可，3 号端子悬空。

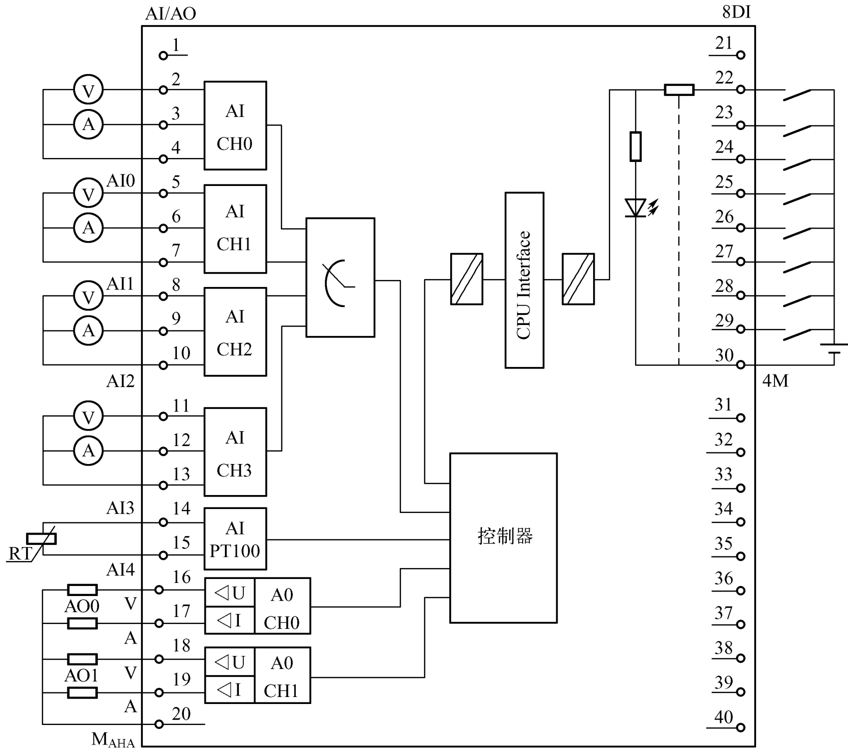


图 2-15　CPU314C－2DP 的模拟 I/O 和数字量输入的接线

14 号和 15 号端子上只能接入热电阻 Pt100。

AO0 和 AO1 是两个模拟输出通道，每个模拟输出通道也是都可输出模拟电压信号和电流信号，但二者只能选择其一，不能同时输出，例如在 AO0 中要输出电压信号，那么只要将电压信号连接在 16 号端子和 20 号端子之间即可，17 号端子悬空。读者可能已经发现这里只谈到接线端子，却未提到输入/输出地址，有关地址将在后续章节讲解。

## 2.3 S7 -300 数字量模块及其接线

S7 -300 有多种型号的数字量输入/输出（I/O）模块供选择。以下将介绍数字量输入模块 SM321、数字量输出模块 SM322 和数字量输入/输出模块 SM323。

### 2.3.1 数字量输入模块 SM321

**1. 数字量输入模块 SM321 的工作原理**

数字量模块用于采集现场过程的数字信号，有的模块采集直流电信号，有的模块则可采集交流信号，并把它转化成 PLC 内部的信号电平。对于现场输入元件，仅要求提供开关触点即可。输入信号进入模块后，一般经过光电隔离、滤波，然后送到输入缓冲寄存器等到 CPU 采样，采样信号经过背板总线进入输入映像区。SM321 模块外形如图 2-16 所示。



图 2-16 SM321 模块外形

用于采集直流信号的模块称为直流输入模块，如图 2-17 所示，一般的输入电压为 DC 24 V。用于采集交流信号的模块称为交流输入模块，如图 2-18 所示，一般的输入电压为 AC 120 V 或者 AC 230 V。

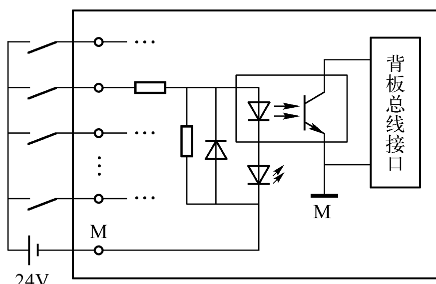



图 2-17 直流数字量模块

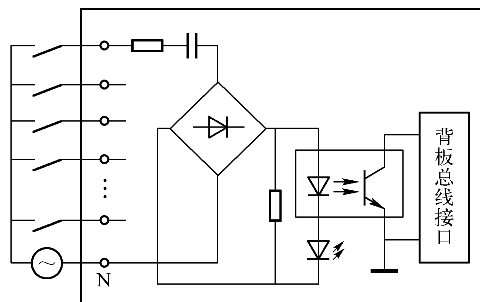



图 2-18 交流数字量模块

**2. 数字量输入模块 SM321 的接线**

数字量输入模块 SM321 有多种型号，以下介绍三种有代表性的输入模块的接线。

直流数字量输入模块多为 PNP 输入，只有个别型号为 NPN 输入，如图 2-19 所示的 SM321 模块是 16 点直流 PNP 输入模块，读者在判断是否为 PNP 型时，只要判断输入有效信号是否是高电平即可。很明显在图 2-10 中，当开关闭合时，输入的是高电平，因此是 PNP 输入。输入开关的一端与模块的端子相连，而另一端则与同一个电源的 24 V 相连。

PNP 型数字量输入模块 SM321 的接线与紧凑型 CPU 的数字量的输入类似。

如图 2-20 所示的 SM321 模块是 16 点直流 NPN 输入模块，其有效输入信号是低电平。

输入开关的一端与模块的端子相连，而另一端则与同一个电源的 0 V 相连。在接线时要特别注意。

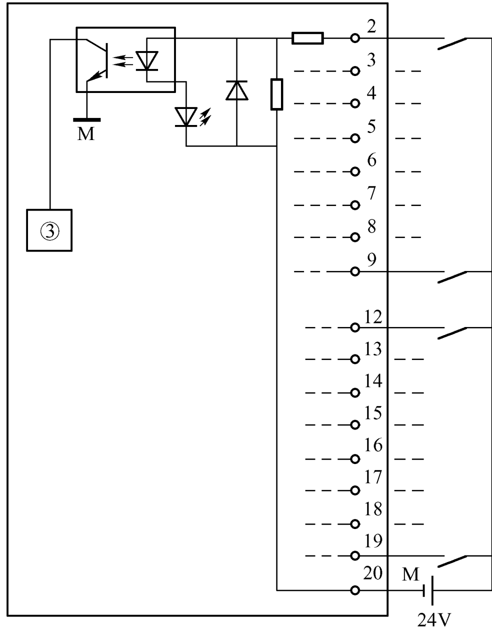



图 2-19　直流数字量输入模块接线图（PNP）

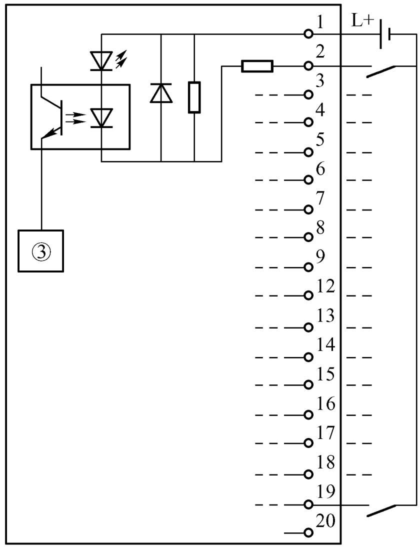



图 2-20　直流数字量输入模块接线图（NPN）

**【关键点】** 本书的 PNP 输入或者 NPN 输入的对象是针对传感器来描述的，如果对象是针对 PLC 描述则刚好与本书不同，则本书的“PNP 和 NPN 输入”将改为“NPN 和 PNP 输入”，这点要特别注意。

交流数字量输入模块的接线如图 2-21 所示。交流电源零线（或者火线）与 1N、2N、3N 和 4N 相连，交流电源火线（或者零线）与输入开关一端相连，而开关的另一端与模块的输入接线端子相连。输入开关可以是按钮、交流接近开关等。

交流模块一般用于强干扰场合。

**【关键点】** 西门子的直流输入模块一般是 PNP 型输入，因此若系统使用接近开关时，最好选用 PNP 接近开关。交流输入模块则不存在此问题，只要选用交流接近开关即可。

## 2.3.2　数字量输出模块 SM322

### 1. 数字量输出模块 SM322 的工作原理

数字量输出模块将 PLC 内部的电平信号转换成外部过程需要的电平信号，同时具有隔离和信号放大的作用，可直接用于驱动电磁阀、接触器、继电器和小功率器件等。

输出模块有晶体管输出（直流输出）、晶闸管输出（交流输出）和继电器输出（交直流输出）三种类型。晶体管输出模块只能驱动直流负载，晶闸管输出模块只能驱动交流负载，而继电器输出模块可以驱动交流和直流负载。从响应速度来看，晶体管输出速度最快，继电器输出响应速度最慢，但从使用的灵活性来看，继电器输出最为灵活。三种模块的内部结构如图 2-22、图 2-23 和图 2-24 所示。

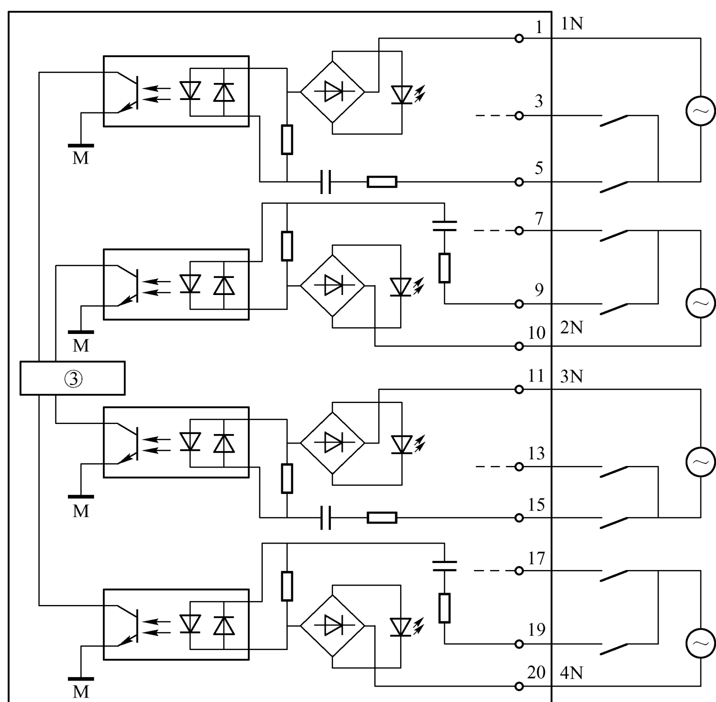


图 2-21　交流数字量输入模块接线图

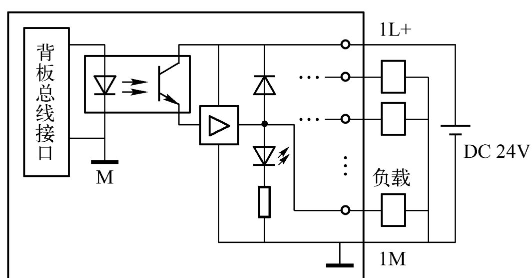


图 2-22　数字量输出模块（晶体管输出）

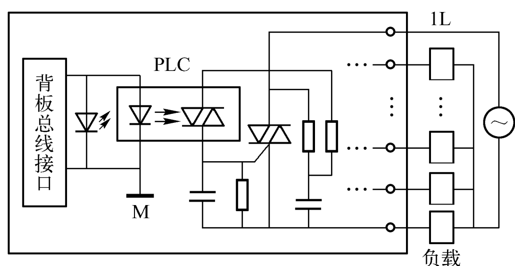


图 2-23　数字量输出模块（晶闸管输出）

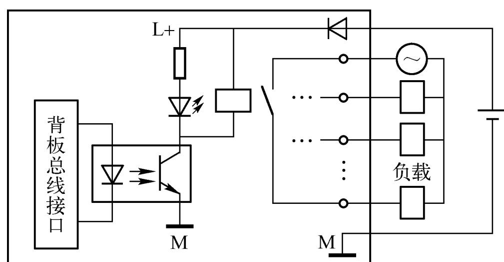


图 2-24　数字量输出模块（继电器输出）

**2. 数字量输出模块 SM322 的接线**

直流数字量输出模块接线如图 2-25 所示，直流数字量输出模块是 PNP 输出（有效输出信号为高电平），负载的一端与模块的接线端子（如 3）相连，负载的另一端与电源的 0 V

相连。1L＋和2L＋与电源的DC ＋24 V相连，1 M和2 M与电源的0 V相连。

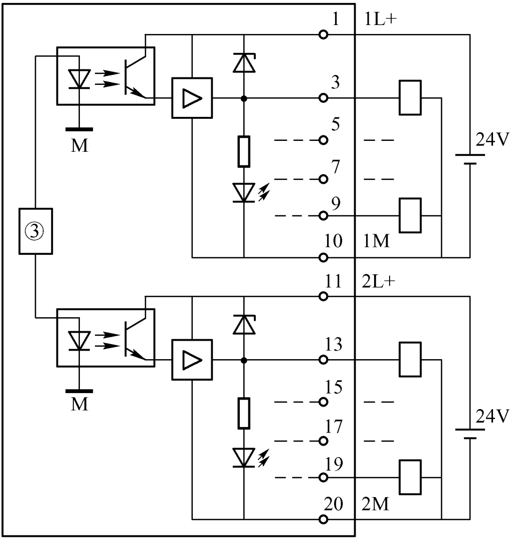



图2-25　直流数字量输出模块接线图

交流数字量输出模块接线如图2-26所示，交流数字量输出模块使用相对较少，其响应速度介于直流数字量输出模块和继电器数字量输出模块之间。

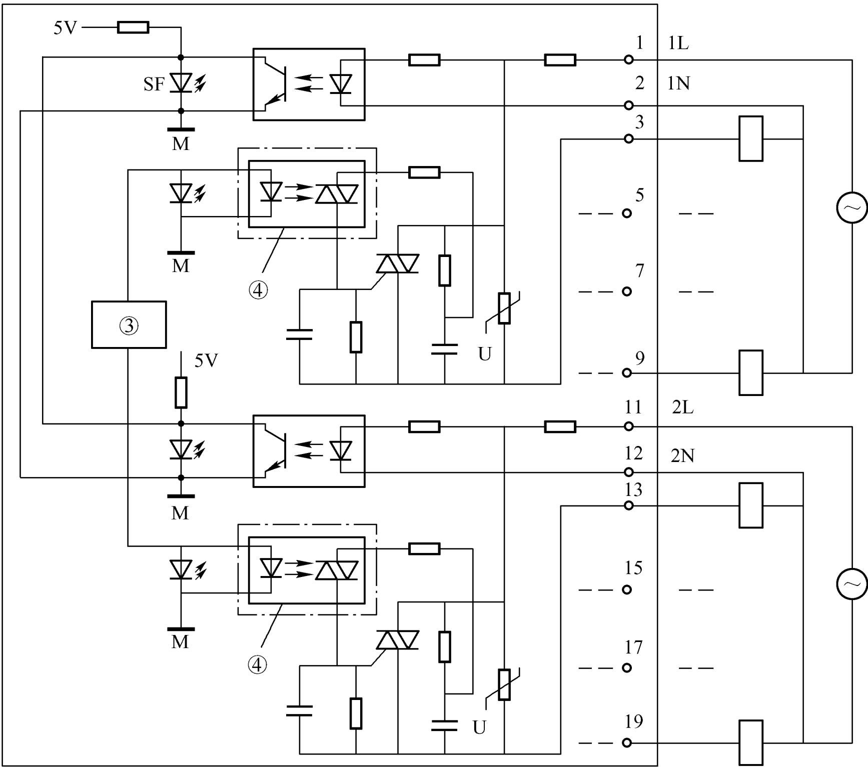



图2-26　交流数字量输出模块接线图

继电器数字量输出模块接线如图 2-27 所示，㊂表示负载的交流式直流电源。

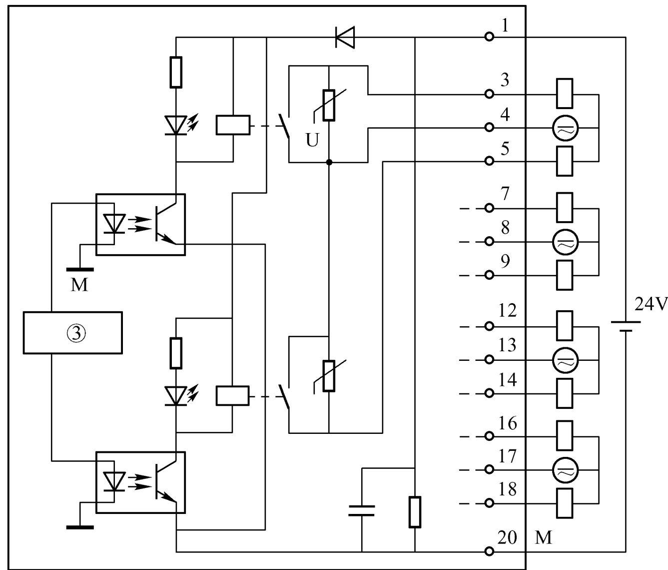



图 2-27　继电器数字量输出模块接线图

SM323 是 S7－300 的数字量输入/输出模块上兼有数字输入和数字量输出，其接线与数字输入模块和数字量输出模块类似，在此不再赘述。

## 2.4　S7－300 模拟量模块及接线

S7－300 系列 PLC 用于模拟量输入（A－D 转换）/输出（D－A 转换）的特殊功能模块有 SM331 模拟量输入模块、SM332 模拟量输出模块、SM334 模拟量输入/输出模块和 SM335 快速模拟量输入/输出混合模块四类。每类中根据输入/输出通道数、分辨率以及连接传感器的不同，又分为不同的规格。由于规格较多，限于篇幅，以下只介绍有代表性的模块。

### 2.4.1　模拟量输入模块 SM331 连接

S7－300 系列 PLC 单独用于模拟量输入的模块 SM331 目前有十几种规格，这些规格依靠订货号 6ES7 331-×××××－0AB0 的中间的 5 位（×××××）进行区分。模拟量输入模块通过总线连接器与 CPU 或者扩展单元相连。

如图 2-28 所示，6ES7 331-7HF01-0AB0 模块是 8 通道 14 位 A－D 转换模块，简称 AI8×14。使用时要注意如下事项：

1）为了提高模拟量输入的可靠性，减少干扰，所有的连线均应使用屏蔽双绞线，屏蔽层要“双端接地”，接线长度要小于 200 m。

2）为了缩短扫描时间，在模块的硬件配置设定时，应将模块全部没有使用的通道设定为“禁止”状态。

3）当选择 1～5 V 模拟电压输入时，如通道组只用 1 个通道，应将未使用的与同组已经使用的输入进行并联。

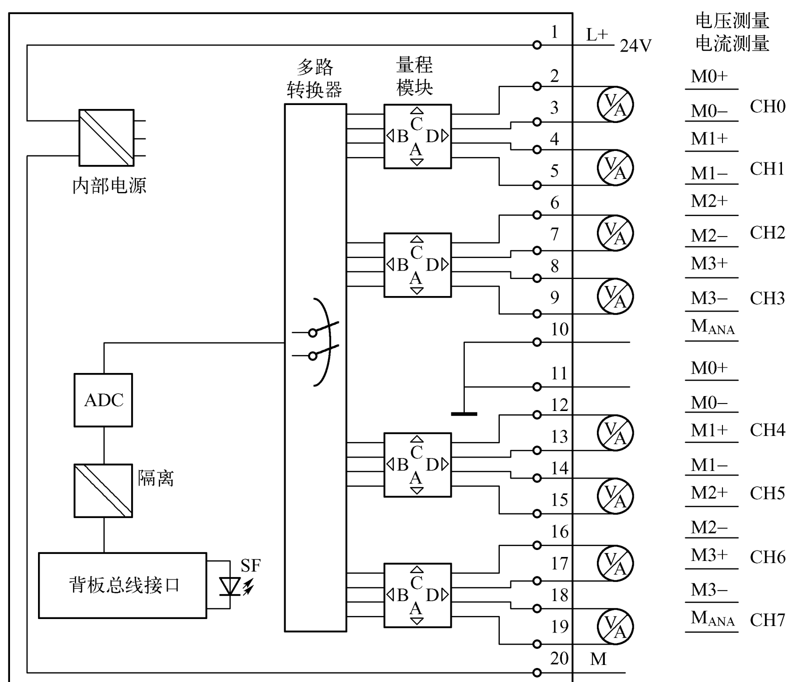



图 2-28　模拟量输入模块（6ES7 331-7HF01-0AB0）接线图

4）当选择 -1～1 V 模拟电压输入时，应将未使用的 Mn + 与 Mn - 短接，并连接到参考电位 MANA（接线端子 11）上。

5）对于两线式模拟电流输入，应将没有使用的输入端开路（不能使用模块的诊断功能），或者在输入端接 1.5～3.3 kΩ 的电阻（能使用模块的诊断功能）。

6）对于 4 线式模拟电流输入，如通道组只用 1 个通道，应将未使用的与已经使用的输入进行串联。

7）此模块可以连接模拟电压和电流信号输入，但不能连接热电阻或者热电偶。

8）该模块只有一个 ADC，所以一次只能转换一个通道的数据，其他通道需要多路开关的切换依次进行，转换后的数据存储到各自的地址中。

如图 2-29 所示，6ES7 331-7PF01-0AB0 模块是 8 通道热电阻输入模块，简称 AI8 × RTD。使用时要注意如下事项：

① 此模块只能用于热电阻输入，4 个通道使用了 4 个 ADC，一次可以完成 4 通道 A - D 转换。

② 遵守 6ES7 331-7HF01-0AB0 模块的 1）～6）项。

③ 未使用的通道应连接标准电阻作为“模拟”输入信号。

④ IC0 + 和 IC0 - （还有 IC1 + 和 IC2- ～IC7 + 和 IC7 - ）是模块提供的电流源，不可以短接，否则会短路。

如图 2-30 所示，6ES7 331-7PF11-0AB0 模块是 8 通道热电偶输入模块，简称 AI8 × TC。使用时要注意如下事项：

① 此模块只能用于热电偶输入，4 个通道使用了 4 个 ADC，一次可以完成 4 通道 A - D 转换。

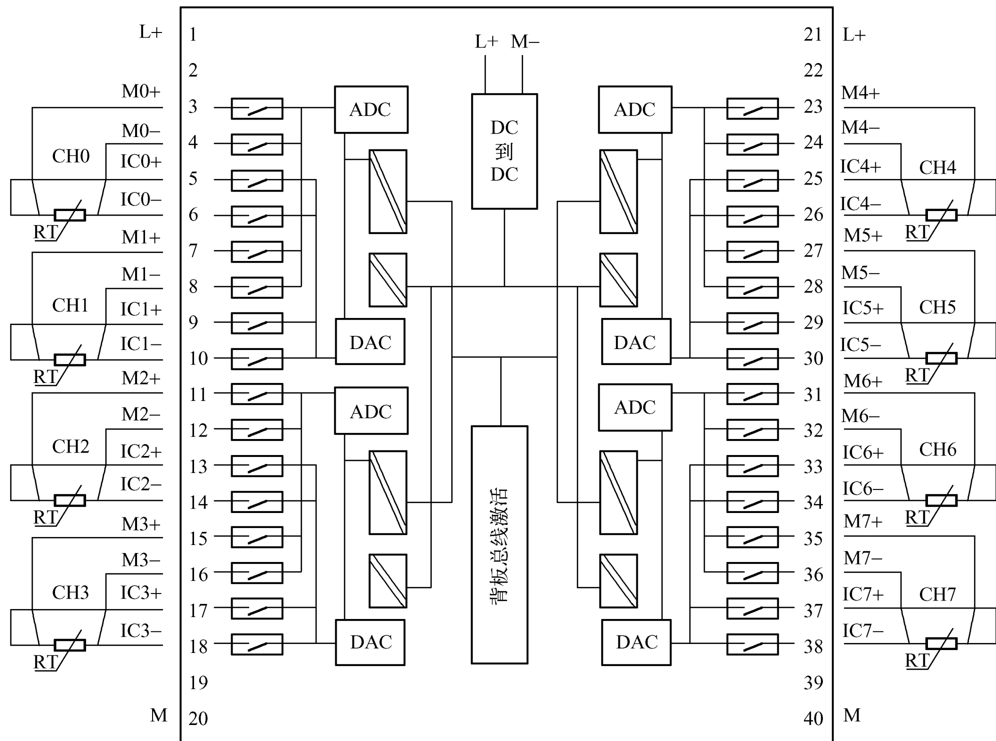



图 2-29　模拟量输入模块（6ES7 331-7PF01-0AB0）接线图

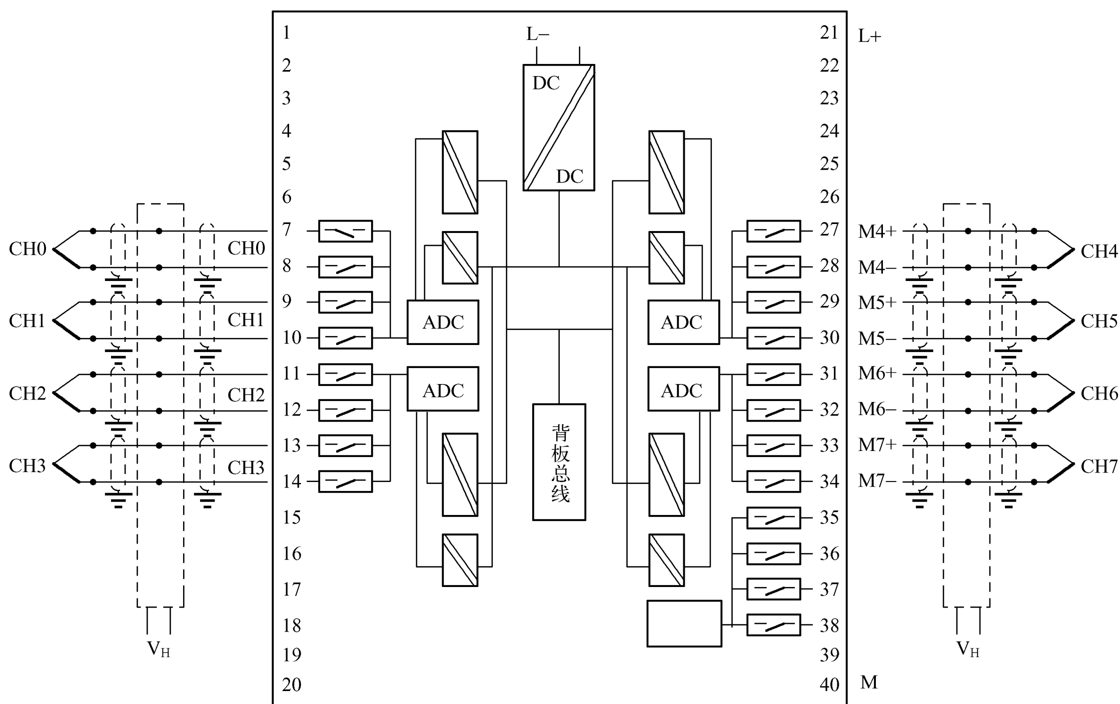



图 2-30　模拟量输入模块（6ES7 331-7PF11-0AB0）接线图

② 遵守 6ES7331-7HF01-0AB0 模块的 1)~6)项。

③ 未使用的通道应将输入端“+”和输入端“-”短接。

使用模拟量模块时，有时需要设置量程卡，这点很重要。此内容在后续章节介绍。

### 2.4.2 模拟量输出模块 SM332 连接

S7-300 系列 PLC 单独用于模拟量输出的模块 SM332 目前有多种规格，分辨率为 12 位和 16 位，模拟输出通道有 2、4、8 共 3 种。模拟量输入模块通过总线连接器与 CPU 或者扩展单元相连，数据通过内部总线传送。

如图 2-31 所示，6ES7 332-5HD01-0AB0 模块是 8 通道模拟量输出模块。使用时要注意如下事项：

1) 当模拟量输出为电压信号时，可采用四线式和两线式两种连接方式。当采用四线式连接时，应将负载连接到 QVn 与 MANA 端，S+和 S-用于连接负载检测端，采用这种形式可以提高模拟量输出准确度。当采用两线式连接时，S+和 S-开路。

2) 当采用电流输出时，应将负载连接到 QVn 与 MANA 端，S+和 S-开路。

3) 为了提高模拟量输入的可靠性，减少干扰，所有的连线均应使用屏蔽双绞线，对于四线式连接时，QVn/S+与 MANA/S-分别使用一对双绞线。

4) 模拟量输出的最大连接距离为 200 m。

**【例 2-4】** 某系统由 CPU313C、SM332 和 MM440 变频器，电动机有起停控制，并对变频器进行模拟量速度给定，请进行正确接线。

**解**：接线图如图 2-32 所示。

**【例 2-5】** 现场一台 CPU315-2DP 采用 PID 输出调节变频器的频率，发现 PID 输出到 100%（也就是 AO 模块输出最大 20 mA 电流信号），但是变频器的频率没有到最大的 50 Hz，测量电流只有 16 mA。分析是什么原因？

**解**：查找问题，发现 AO 模块后面有隔离器，去掉隔离器，AO 模块直接输出到变频器，输出信号正常，问题解决。

### 2.4.3 模拟量输入/输出模块 SM334 连接

S7-300 系列 PLC 同时用于模拟量输入和模拟量输出的模块是 SM334 和 SM335。SM334 是普通的 A-D、D-A 模块，目前有两种规格；SM335 是高速 A-D、D-A 转换模块，目前只有一种规格。模拟量输入/输出模块通过总线连接器与 CPU 或者扩展单元相连，数据通过内部总线传送。

如图 2-33 所示，6ES7 334-0CE01-0AA0 模块是 4 通道模拟量输入和两通道模拟量输出模块。它实际上就是模拟量输入和模拟量输出模块的合并，因此其使用时要注意的事项与模拟量输入和模拟量输出模块的相同，在此不再赘述。

SM334 模块的地址分配见表 2-3。

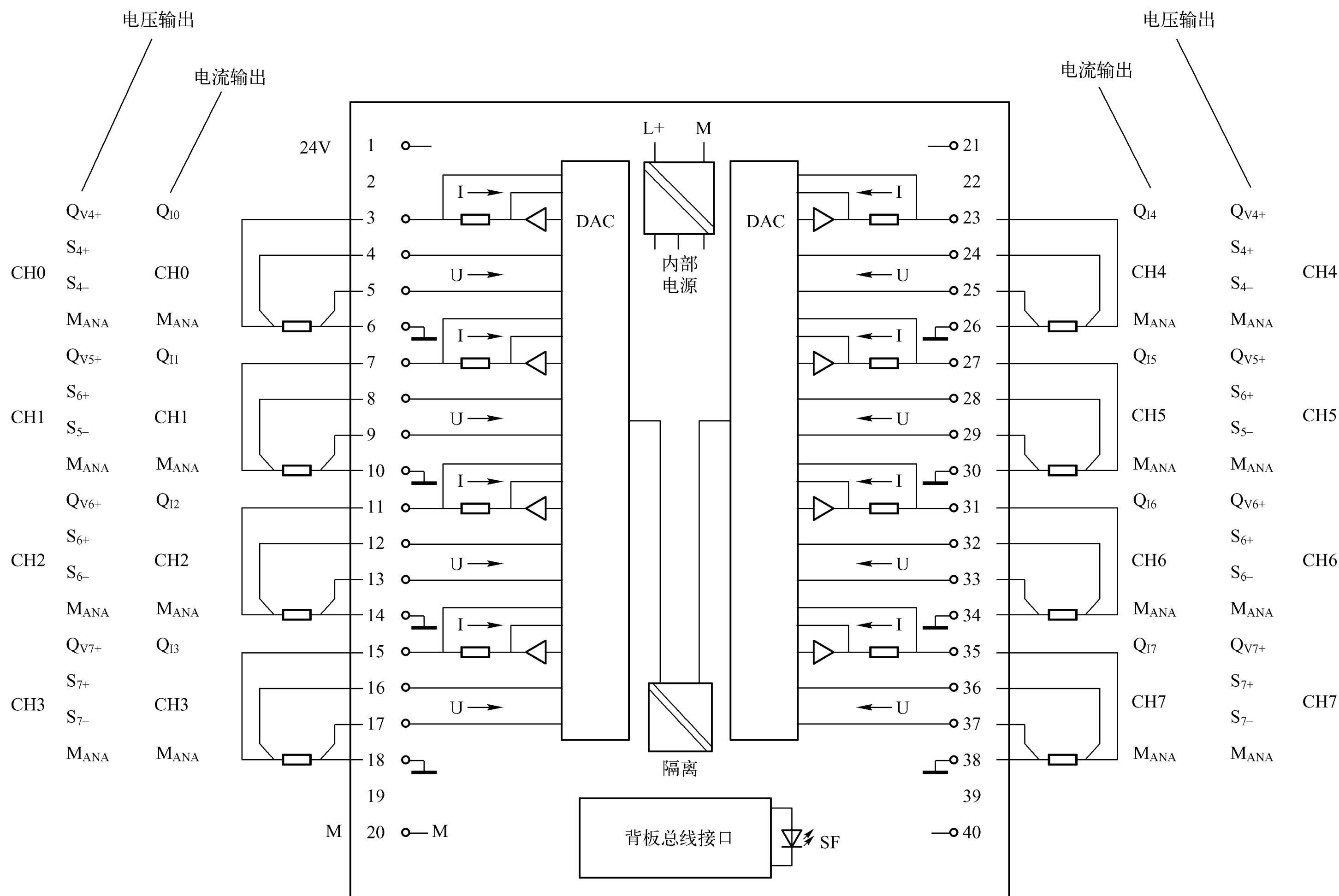



图2–31 模拟量输出模块（6ES7 332–5HD01–0AB0）接线图

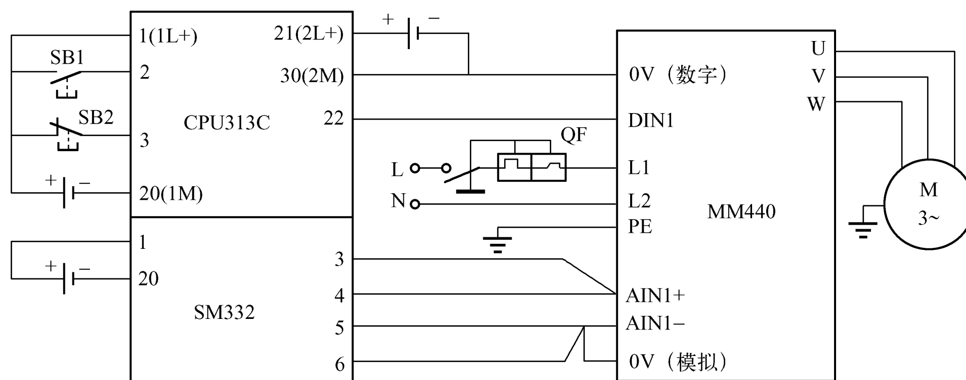



图 2-32　接线图

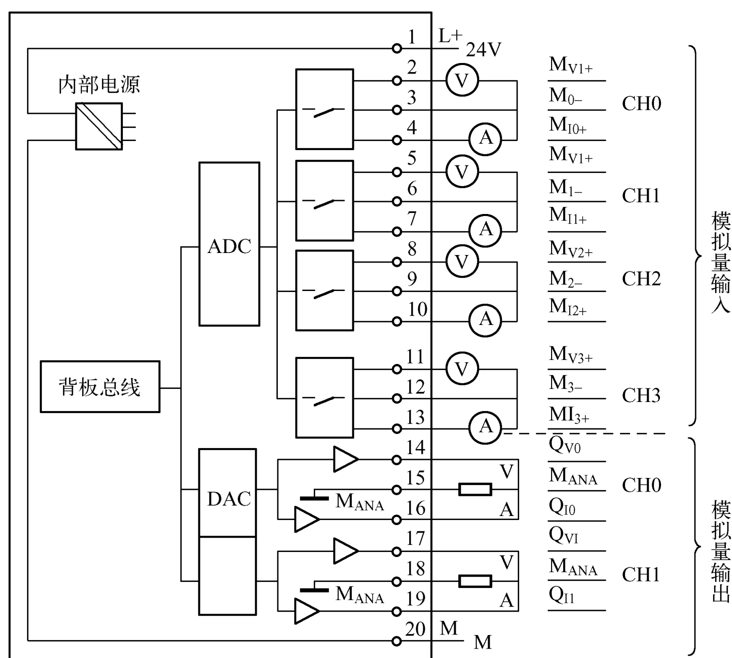



图 2-33　模拟量输入/输出模块（6ES7 334－0CE01－0AA0）接线图

**表 2-3　SM334 模块的地址**

| 通　　道 | 地　　址 |
|---|---|
| 输入通道 0 | 模块的起始地址 |
| 输入通道 1 | 模块的起始地址 +2B 的地址偏移量 |
| 输入通道 2 | 模块的起始地址 +4B 的地址偏移量 |
| 输入通道 3 | 模块的起始地址 +6B 的地址偏移量 |
| 输出通道 0 | 模块的起始地址 |
| 输出通道 1 | 模块的起始地址 +2B 的地址偏移量 |

【例 2-6】某设备的控制器为 CPU315-2DP，有 3 个按钮输入，数字量输入模块为 6ES7 321-1BH02-0AA0，有 4 个电磁阀（两只为 DC 24 V，两只为 AC 220 V），数字量输出模块为 6ES7 322-1HF01-0AA0，3 路电流模拟量输入，模拟量输入模块为 6ES7 331-7HF01-0AB0，3 路电压模拟量输出，模拟量输出模块为 6ES7 332-5HD01-0AB0，请设计接线图。

**解**：CPU 和扩展模块以及扩展模块之间通过总线连接器连接；数字量输入模块是 PNP 输入；数字量输出模块是继电器输出。由于电磁阀的额定电压有 DC 24 V 和 AC 220 V 两种，所以一般选用继电器输出模块，而且不同额定电压的电磁阀不连接到一组；模拟量输出信号为电压信号，采用四线式连接（当然也可以采用两线式连接），接线图如图 2-34所示。

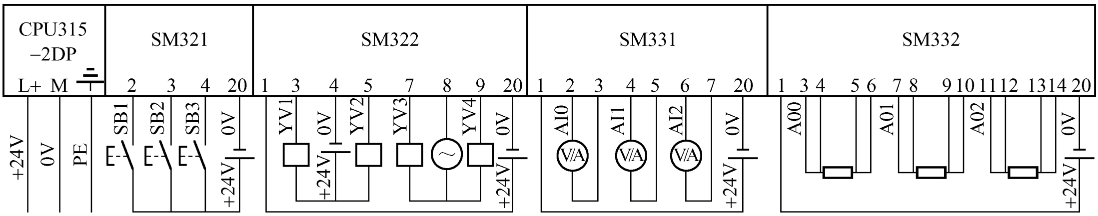



图 2-34　接线图

## 2.5　S7-300 的其他模块简介

### 2.5.1　S7-300 系列 PLC 的通信处理模块

通信处理模块为 PLC 接入 PROFIBUS 现场总线、工业以太网和串行通信等提供了极大的便利。通过集成在 STEP7 中的参数化处理工具可进行简便的参数设置。通信处理模块很常用，代号以“CP”开头。主要分为四大类，以下分别进行介绍。

**1. PROFIBUS 通信模块（CP342-5）**

CP342-5 是用于 PROFIBUS 总线系统的 SIMATIC S7-300 和 SIMATIC C7 的通信模块。CP342-5 减轻了 CPU 的通信任务。它使用的通信协议有：PROFIBUS-DP、PG/OP 通信及 S7 通信等。使用 CP342-5 通信处理模块传输速度可达 9.6 kbit/s ~ 12 Mbit/s。

CP342-5 是用于传输电信号，CP342-5 FO 是用于传输光纤信号。

PROFIBUS 通信模块（CP342-5）很常用。

**2. 以太网通信模块（CP343-1）**

CP343-1 在工业以太网上独立处理数据通信，主要用于操作员之间的连接。该模块有自身的处理器，符合 OSI 模型，支持的协议有 ISO、TCP/IP、S7 通信和 PG/OP 通信等。以太网模块目前有 CP343-1 Lean、CP343-1、CP343-1 IT 和 CP343-1 PN 等类型，各有特色。其中 CP343-1 Lean 最低端，只能作服务器端，而不能作客户端。以太网通信模块很常用。

**3. 点对点通信模块**

点对点通信模块使用量相对较少。以下介绍两类。

① CP340。CP340 通过点对点连接进行串行通信的经济型解决方案。它具有 3 种不同物

理接口，分别是 RS－232C（V.24）、20 mA（TTY）和 RS－422/RS－485（X.27）。CP340 执行的协议有 ASCII 、3964（R）（不适于 RS－485）和打印机驱动程序。

② CP341。CP341 用于执行强大的点到点高速串行通信。具有不同物理特性的 3 个型号，分别是 RS－232C（V.24）、20 mA（TTY）和 RS－422/RS－485（X.27）。CP341 执行的协议有 ASCII、3964（R）、RK 512 和客户协议（可装载）。

**4. AS－Interface 通信模块（CP343-2）**

CP343-2 是用于 SIMATIC S7－300 PLC 和 ET 200M 分布式 I/O 的 AS－Interface 主站。CP343-2 最多可连接 62 个 AS－Interface 从站，并且可集成模拟值传送（遵循扩展 AS－Interface 规范 V2.1）。支持所有 AS－Interface 主站，符合扩展 AS－Interface 接口技术规范 V2.1 。这类模块应用相对较少。

## 2.5.2 S7－300 系列 PLC 的功能模块

**1. 计数器模块**

模块的计数器均为 0～32 位或 ±31 位加减计数器，可以判断脉冲的方向，模块给编码器供电。达到比较值时发出中断。可以 2 倍频和 4 倍频计数。有集成的 DI/DO。

FM 350－1 是单通道计数器模块，可以检测最高达 500kHz 的脉冲，有连续计数、单向计数和循环计数 3 种工作模式。FM 350－2 和 CM 35 都是 8 通道智能型计数器模块。

其功能是除了在高速计数外，还可以在同步模式测量模式下工作。模块在频率测量的基础上增加了转速测量和周期测量的功能。此外在诊断中断的内容方面，增加了外部电压故障、DC 5 V 故障、RAM 出错和 A/B/N 输入错误等检测功能。

**2. 位置控制与位置检测模块**

FM351 双通道定位模块用于控制变级调速电动机或变频器。FM353 是步进电动机定位模块。FM354 是伺服电动机定位模块。FM357 可以用于最多 4 个插补轴的协同定位。FM352 高速电子凸轮控制器，它有 32 个凸轮轨迹，13 个集成的 DO，采用增量式编码器或绝对式编码器。

SM338 超声波传感器用于检测位置，无磨损、保护等级高且准确度稳定不变。

**3. 闭环控制模块**

FM355 闭环控制模块有 4 个闭环控制通道，有自优化温度控制算法和 PID 算法。

功能模块的使用比较复杂，接线也相对复杂，请读者参考相关手册，在此不再赘述。

## 2.5.3 ET－200 的模块

ET－200 的模块是西门子基于 PROFIBUS 或者 PROFINET 的分布式控制模块，应用非常广泛。它有 5 大类，分别如下。

**1. ET200M 模块**

ET200M 模块是一种多通道模块化的分布式 I/O 系统，可以使用 S7－300 的模块。它要安装在控制柜内，其外形如图 2－35 所示。

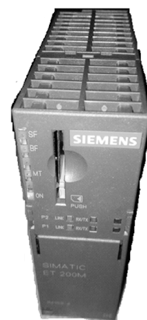


图 2-35　ET200M 外形

**2. ET200S 模块**

ET200S 模块是一种多功能的按位模块化的分布式 I/O 系统，

体积较小，安装在控制柜内。

**3. ET200isp 模块**

ET200isp 模块是一种本质安全型模块化的分布式 I/O 系统，要安装在控制柜内。

**4. ET200pro 模块**

ET200pro 模块的防护等级为 IP65/67，是模块化的分布式 I/O 系统，可直接安装在工控现场，如机架上，不用安装在控制柜中。

**5. ET200eco 模块**

ET200eco 模块是可直接安装在工控现场，如机架上，不用安装在控制柜中。它是经济型的模块化的分布式 I/O 系统。

## 2.5.4 电源模块

PS 307 电源模块将 AC 120/230 V 电压转换为 DC 24 V 电压，为 S7-300、传感器和执行器供电，实际上就是开关电源。输出电流有 2 A、5 A 或 10 A 等规格。电源模块安装在导轨上的插槽 1。电源模块外形如图 2-36 所示。S7-300 的浮动参考电位如图 2-37 所示。

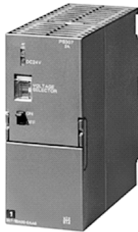


图 2-36 PS 电源模块（配 S7-300）外形

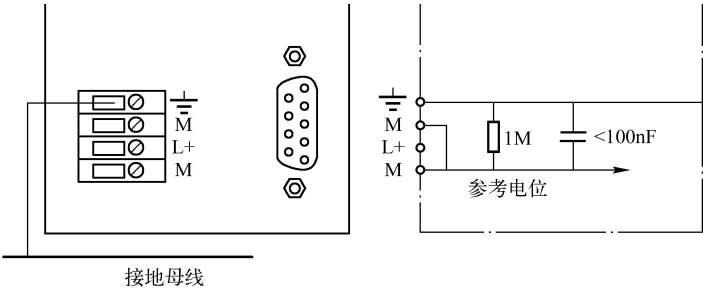



图 2-37 S7-300 的浮动参考电位

某些大型工厂（例如化工厂和发电厂）为了监视对地的短路电流，可能采用浮动参考电位，可以将 M 点与接地点之间的短接片去掉。

## 2.5.5 接口模块

接口模块（IM）用于多机架配置时连接主机架（CR）和扩展机架（ER），其外形如图 2-38 所示。S7-300 通过分布式的主机架和连接的扩展机架（最多可连接 3 个扩展机架），主机架最多安装 11 个模块、1 个电源模块、1 个 CPU 模块、1 个接口模块和 8 个扩展模块。而扩展机架最多安装 10 个模块、1 个电源模块、1 个接口模块和 8 个扩展模块，不能安装 CPU 模块。由于 CPU 最多连接 4 个机架，因此最多可以操作最多 32 个扩展模块。



图 2-38 接口模块外形

对于 S7-300，IM365 接口模块是经济型的，只有一个扩展机架时，中央机架和扩展机架使用一对 IM365，且优先选用 IM365；当扩展机架超过两个（含两个）时，中央机架用 IM360，扩展机架用 IM361。S7-300 的扩展能力示意如图 2-39 所示。

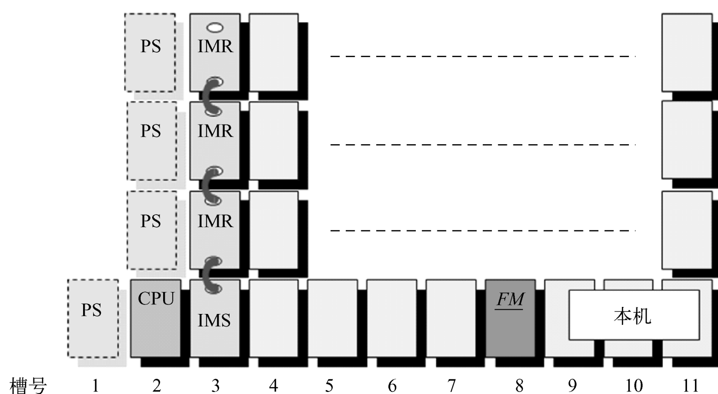



图 2-39　S7 -300 的扩展能力示意图

S7 -300 模块组成的完整的系统如图 2-40 所示。

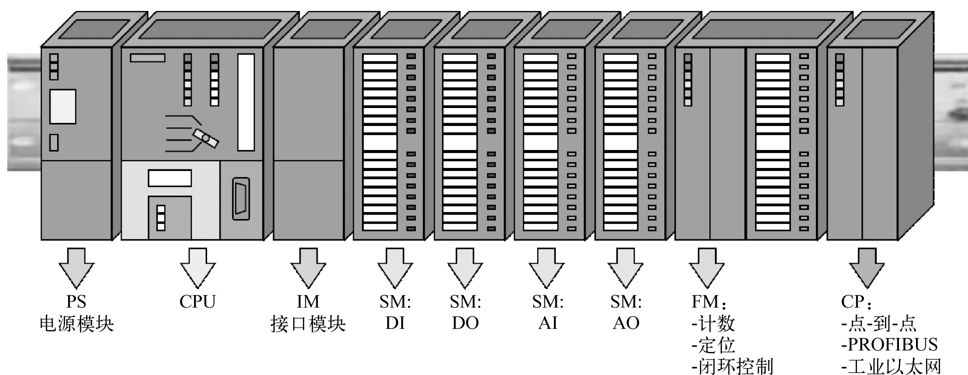



图 2-40　S7 -300 模块组成的完整的系统

### 2.5.6　导轨

导轨（RACK）是安装 S7 -300 各类模块的机架，它是特制的铝型材异形板，其标准长度有 160 mm、482 mm、530 mm、830 mm 和 2000 mm，可以根据实际选用。S7 -300 的导轨上无背板总线。但 S7 -400 有背板总线，不可缺少。导轨的外形如图 2-41 所示。S7 -300 的导轨不是 35 mm 的 DIN 导轨。




图 2-41　S7 -300 的导轨外形

## 2.6　S7 - 400 PLC 概述

### 2.6.1　S7 -400 的结构

SIMATIC S7 -400 是用于中、高档性能范围的 PLC，其尺寸比 S7 -300 大。因其采用模块化及无风扇的设计、坚固耐用、容易扩展和广泛的通信能力、容易实现的分布式结构以及

用户友好的操作等优点，使 SIMATIC S7 –400 成为中、高档性能控制领域中首选的理想解决方案。S7 –400 系统的配置比较灵活，不同的配置可以选用不同的模块。S7 –400 常用的模块有电源模块、CPU、接口模块 IM、数字量输入模块 DI、数字量输出模块 DO、模拟量输入模块 AI、模拟量输出模块 AO、功能模块 FM 和通行处理模块 CP 等。

### 2.6.2　S7 –400 的特点

与 S7 –300 PLC 相比，S7 –400 PLC 的体积更大，性能更加卓越，其特点如下：

1）运行速度高，CPU 416 执行一条二进制指令只要 0.08 μs。

2）存储器容量大，例如 CPU 417 –4 的 RAM 可以扩展到 16 MB，装载存储器（EEPROM 或 RAM）可以扩展到 64 MB。

3）I/O 扩展功能强，可以扩展 21 个机架，CPU 417 –4 最多可以扩展 262 144 个数字量 I/O 点和 16 384 个模拟量 I/O。

4）有极强的通信能力，集成的 MPI 能建立最多 32 个站的简单网络。大多数 CPU 集成有 PROFIBUS – DP 主站接口，用来建立高速的分布式系统，通信速率最高 12 Mbit/s。

5）集成的 HMI 服务，只需要为 HMI 服务定义源和目的地址，自动传送信息。

S7 –400 PLC 的外形如图 2–42 所示。

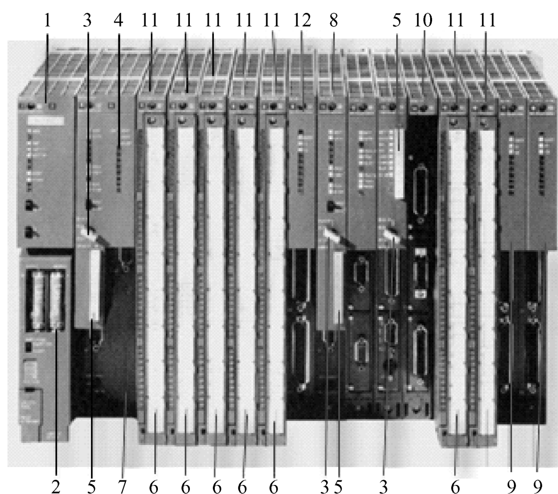



图 2–42　S7 –400 PLC 的外形图

1—电源模板　2—后备电池　3—钥匙开关　4—状态和故障 LED　5—存储器卡
6—有标签区的前连接器　7—CPU 1　8—CPU 2　9—CP 接口　10—IM 接口模板　11—I/O 模板

## 2.7　S7 –400 CPU 模块

CPU 模块是控制系统的核心，负责系统的中央控制、存储并执行程序以及实现通信功能。

### 2.7.1 S7-400 CPU 模块的基本结构

S7-400 的 CPU 分别适用于不同等级的控制要求。不同的 CPU 的面板差别较大。

**1. S7-400 CPU 模块的指示灯与模式选择开关**

S7-400 CPU 模块上指示灯的功能见表 2-4。这些指示灯在调试和故障诊断时很有用。

**表 2-4 S7-400 CPU 模块上指示灯的功能**

| 序 号 | 指 示 灯 | 颜 色 | 含 义 |
|---|---|---|---|
| 1 | INTF | 红色 | 内部故障，如用户程序错误 |
| 2 | EXTF | 红色 | 外部故障，如模块故障，电池电压低 |
| 3 | FRCE | 黄色 | 输入或者输出使用了强制功能 |
| 4 | RUN | 绿色 | 运行状态模式 |
| 5 | STOP | 黄色 | 停止状态模式 |
| 6 | BUS1F | 红色 | MPI/PROFIBUS-DP 接口 1 的总线故障 |
| 7 | BUS2F | 红色 | MPI/PROFIBUS-DP 接口 2 的总线故障 |
| 8 | MSTR | 黄色 | CPU 运行 |
| 9 | REDF | 红色 | 冗余错误 |
| 10 | RACK0 | 黄色 | CPU 在机架 0 中 |
| 11 | RACK1 | 黄色 | CPU 在机架 1 中 |
| 12 | IFM1F | 红色 | 接口子模板 1 故障 |
| 13 | IFM2F | 红色 | 接口子模板 2 故障 |

S7-400 CPU 模块的模式选择开关的外形和使用方法与 S7-300 的完全相同，开关的含义也与 S7-300 的完全相同。

**2. 存储卡**

S7-400 的存储卡外形如图 2-43 所示，其外形尺寸比 S7-300 的存储卡大了很多。S7-400 的存储卡包含 RAM 卡和 FLASH 卡。用 RAM 卡可以扩展 CPU 装载存储器的容量，掉电后内容不能保存。FLASH 卡存储用于程序和数据，即使在没有后备电池的情况下，其内容也不会丢失。可以在编程器或 CPU 上编写 FLASH 卡的内容。FLASH 卡也可以扩展 CPU 装载存储区的容量。




图 2-43 S7-400 的存储卡外形

**3. 后备电池**

S7-400 需要一块或者两块后备电池（新型的 S7-300 不需要后备电池），通过背板总线备份 CPU 和可编程控制模块中的参数设置和 RAM 中的存储内容。

### 2.7.2 S7-400 CPU 模块的分类和技术规范

**1. S7-400 CPU 模块的分类**

S7-400 CPU 有多款 CPU，以下分别进行介绍。

（1）CPU 412-1 和 CPU 412-2

CPU 412-1 和 CPU 412-2 用于中等性能范围的小型控制系统。CPU 412-1 有 96 KB 的

RAM，最大可扩展 32 KB 的 DI/DO 和 2 KB 的 AI/AO，有一个组合式的 MPI/DP 接口，允许 PROFIBUS - DP 操作。CPU 412-2 有 144 KB 的 RAM，最大可扩展 32 KB 的 DI/DO 和 2 KB 的 AI/AO，有一个组合式的 MPI/DP 接口，允许 PROFIBUS - DP 操作。此外，这两款 CPU 的运行速度和扩展能力也有所不同。

（2）CPU 414 - 2 和 CPU 414 - 3

CPU 414 - 2 和 CPU 414 - 3 适合于中等性能系统的控制。它们满足对程序规模和指令处理速度以及复杂通信的更高要求。CPU 414 - 2 有 256 KB 的 RAM，最大可扩展 64 KB 的 DI/DO 和 4 KB 的 AI/AO，有一个组合式的 MPI/DP 接口和一个 PROFIBUS - DP 接口，允许 PROFIBUS - DP 操作。CPU 414 - 3 有 768 KB 的 RAM，最大可扩展 64 KB 的 DI/DO 和 4 KB 的 AI/AO，有一个组合式的 MPI/DP 接口、一个 PROFIBUS - DP 接口和一个 IF 模块插槽，允许 PROFIBUS - DP 操作。这两款 CPU 功能较强，具有集成 PROFIBUS - DP 接口，能作为主站直接连接到 PROFIBUS - DP 总线。

（3）CPU 416 - 2 和 CPU 416 - 3

CPU 416 - 2 和 CPU 416 - 3 应用于高性能范围中的各种高要求的场合。CPU 416 - 2 有 1.6 MB 的 RAM，最大可扩展 128 KB 的 DI/DO 和 8 KB 的 AI/AO，有一个组合式的 MPI/DP 接口和一个 PROFIBUS - DP 接口。CPU 416 - 3 有 3.2 MB 的 RAM，最大可扩展 128 KB 的 DI/DO 和 8 KB 的 AI/AO，有一个组合式的 MPI/DP 接口、一个 PROFIBUS - DP 接口和一个 IF 模块插槽。这两款 CPU 功能强大，具有集成 PROFIBUS - DP 接口，能作为主站直接连接到 PROFIBUS - DP 总线。

（4）CPU 417 - 4

CPU 417 - 4 适用于更高性能范围的最高要求的场合，在 S7 - 400 CPU 中功能最强大。具有集成 PROFIBUS - DP 接口，能作为主站直接连接到 PROFIBUS - DP 总线。有两个槽适用于 IF 模块（串行）。

（5）CPU 417 - 4H

CPU 417 - 4H 是 S7 - 400H 和 S7 - 400F/FH 中功能最强的 CPU，是硬件冗余 CPU。可配置为容错式 S7 - 400H 系统。连接上 F 运行授权后，可以作为 S7 - 400F/FH 容错自动化系统应用。集成的 PROFIBUS - DP 接口能作为主站直接连接到 PROFIBUS - DP 现场总线。

S7 - 400H 采用“热备用”模式的主动冗余原理，在发生故障时，无扰动地自动切换。两个控制器使用相同的用户程序，接收相同的数据，两个控制器同步地更新内容，任意一个子系统有故障时，另一个承担全部控制任务。

（6）S7 - 400F

S7 - 400F 是安全型自动化系统，出现故障时转为安全状态，并执行中断。S7 - 400FH 是安全及容错自动化系统，如果系统出现故障，生产过程能继续执行。

**2. S7 - 400 CPU 的技术参数**

S7 - 400 CPU 的技术参数是设计选型的重要依据，因此读者在入门学习时，要养成查看技术参数的习惯，以下仅列出最常用的几个型号的 CPU 的技术参数，见表 2-5。

表 2-5　S7 -400 CPU 的技术参数表

| 型　　号 | 412-1 | 413-1 | 414 -2DP |
|---|---|---|---|
| 用于程序/用于数据的集成 RAM 工作存储器 | 144 KB/144 KB | 256 KB/256 KB | 1. 4 MB/1. 4 MB |
| 装载存储器：集成/Flash 存储卡/RAM 存储卡 | 8 KB/48 KB/16 MB | 8 KB/72 KB/16 MB | 8 KB/128 KB/16 MB |
| 位操作执行时间/ns<br>浮点运算指令执行时间/ns | 200<br>1200 | 200<br>1200 | 100<br>600 |
| 位存储器（M）/KB | 4 | 4 | 8 |
| S7 定时器/计数器 | 256/256 | 256/256 | 256/256 |
| 最大数字量 I/O 点数<br>最大模拟量 I/O 点数 | 32 768/32 768<br>2 048/2 048 | 32 768/32 768<br>2 048/2 048 | 65 535/65 535<br>4 096/4 096 |
| 最大局部数据/KB | 8 | 8 | 16 |
| FB 的最大块数<br>FC 的最大块数<br>DB 的最大块数<br>OB 的最大容量/KB | 256<br>256<br>512<br>64 | 256<br>256<br>512<br>64 | 512<br>512<br>1024<br>64 |

# 2. 8　S7 -400 PLC 的机架

## 2. 8. 1　S7 -400 PLC 的机架简介

S7 -400 的机架是安装所有模块的基本框架，这些模块通过背板总线进行交换数据和供电。S7 -400 的机架种类和应用见表 2-6。

表 2-6　S7 -400 的机架种类和应用

| 机　　架 | 插 槽 总 数 | 可 用 总 线 | 可 用 领 域 | 说　　明 |
|---|---|---|---|---|
| UR1 | 18 | I/O 总线<br>通信总线 | CR 或者 ER | 适用于所有的模块类型 |
| UR2 | 9 | | | |
| ER1 | 18 | 受限 I/O 总线 | ER | 适用于 SM、IM 和 PS 模块<br>I/O 总线受以下控制：<br>1）不会响应模块中断。<br>2）不能使用 24 V 供电模块。<br>3）模块不能使用模块的后备电源供电，也不能通过外加给 CPU 或接收 IM 的电压加电 |
| ER2 | 9 | | | |
| CR2 | 18 | 分段 I/O 总线<br>连续通信总线 | 分段 CR | 适用于除 IM 外所有的模块，I/O 总线分两段，占 10 槽和 8 槽 |
| CR3 | 4 | I/O 总线<br>通信总线 | 标准系统 CR | 适用于除 IM 外所有的模块，CPU41X - H 仅限单机操作 |
| UR2-H | 2 ×9 | 分段 I/O 总线<br>连续通信总线 | 为紧凑安装容错型系统细分为 CR 或者 ER | 适用于除 IM 外所有的模块，I/O 总线分两段，各占 9 槽 |

1）机架的数据交换。I/O 总线（P 总线）是机架的并行背部总线，用于 I/O 信号交流，对于信号模块的过程数据也通过 I/O 总线进行。通信总线（C 总线）是机架背部串行总线，用于快速交换 I/O 信号相关的大量数据。除 ER1 和 ER2 外，其他的机架只有一条通信总线。

2）机架的供电。通过背部总线和基本连接器，由安装在机架最左侧的电源模块为机架上的模块提供所需的工作电压（5 V 用于逻辑控制，24 V 用于接口模块）。对于本地连接，

还可以通过 IM460－1/IM461-1 接口模块为 ER 供电。IM460－1 有两个接口，每个接口最多可以通过 5 A 的电流，也就是说可以为本地连接中的每个 ER 提供 5 A 的电流。

### 2.8.2 UR1 和 UR2 机架（通用机架）

UR1 和 UR2 机架用于装配中央控制器和扩展单元。UR1 和 UR2 机架都有 I/O 总线和通信总线。UR1 最多可容纳 18 个模块，UR2 最多可容纳 9 个模块。UR1 机架如图 2-44 所示。

### 2.8.3 CR2 和 CR3 机架

CR2 机架用于分段式中央机架设计，可安装中央控制器和扩展单元。CR2 有 I/O 总线和通信总线。I/O 总线分为两个局部总线区段，分别为 10 个和 8 个插槽，如图 2-45 所示，I/O 总线分为两个局部总线区段①和②，其中区段①为 10 槽，区段②为 8 槽，③为通信总线。CR2 共 18 槽，可安装一个电源模块和两个 CPU 模块和其他模块。

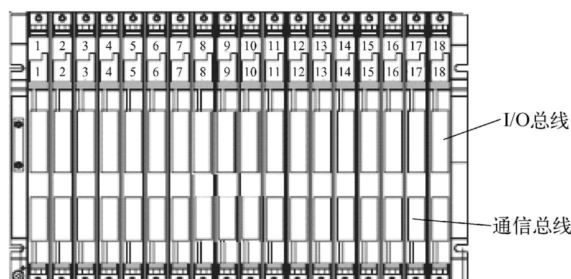



图 2-44　UR1 机架及其总线

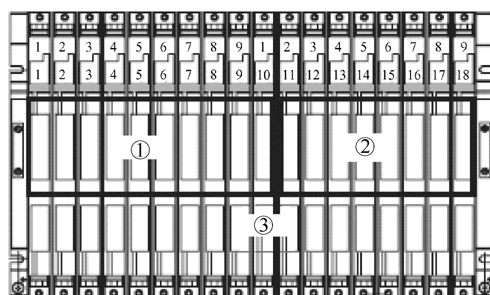



图 2-45　CR2 机架及其总线

CR3 是 4 槽的中央机架，有 I/O 总线和通信总线各一条。

### 2.8.4 UR2-H 机架

UR2-H 机架用于在一个机架上装配两个独立运行的中央控制器或扩展单元。UR2-H 机架实质上代表同一装配导轨上的两个电隔离的 UR2 机架。UR2-H 主要应用于紧凑结构的冗余 S7－400H 系统（同一机架上有两个设备或系统）。

UR2-H 机架及其总线如图 2-46 所示，机架分为①和②两段，每段 9 槽，每段都有各自的电源和 CPU。

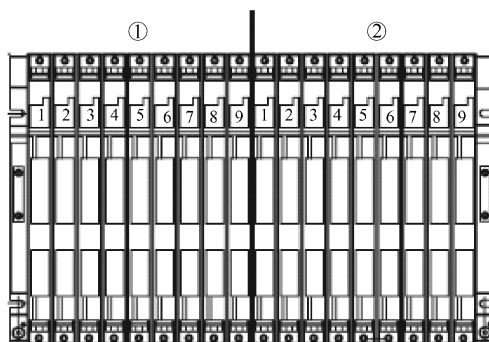



图 2-46　UR2-H 机架及其总线

① 段的 1～9 号槽按顺序分别安装电源模块、CPU 模块和 DI 模块等。

② 段的 1～9 号槽按顺序分别安装电源模块、CPU 模块及 DI 模块等。

注意西门子机架的槽位从 1 开始，而有的 PLC 的机架槽位从 0 开始（如 GE 的 RX3i）。

## 2.9 S7－400 数字量模块及其连接

S7－400 的数字量模块包含数字量输入模块 SM441 和数字量输出模块 SM332，没有混合

模块（S7－300 有混合模块）。

### 2.9.1　数字量输入模块 SM421 及其连接

数字量输入模块将二进制过程信号连接到 S7－400，也就是将按钮、接近开关等数字量连接到 S7－400。从输入电信号的种类分数字量输入模块主要有三种，即直流输入、交流输入和交直流输入。目前数字量输入模块 SM421 有多个规格。以下将介绍有代表性的几种。

数字量输入模块（6ES7 421-1BL01-0AA0）的内部结构和连接如图 2-47 所示，很明显 SM421 模块的内部有光隔离，模块是 DC 24 V，PNP 输入（高电平有效），其连接与 SM321 的 PNP 输入模块的连接类似。

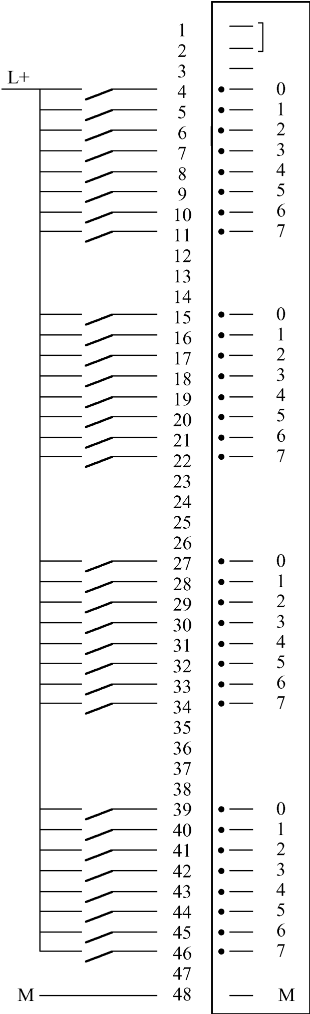


图 2-47　数字量输入模块（6ES7 421-1BL01-0AA0）内部结构和连接

数字量输入模块（6ES7 421-5EH00－0AA0）的内部结构和连接如图 2-48 所示，SM421 模块是交流输入，额定电压为 AC 120 V，可连接普通开关（按钮）和两线接近开关。

数字量输入模块（6ES7 421-1FH00－0AA0）的内部结构和连接如图 2-49 所示，SM421 模块是交直流 16 点输入，额定电压为 VUC 120 V/230 V（VUC 代表交流和直流信号输入均

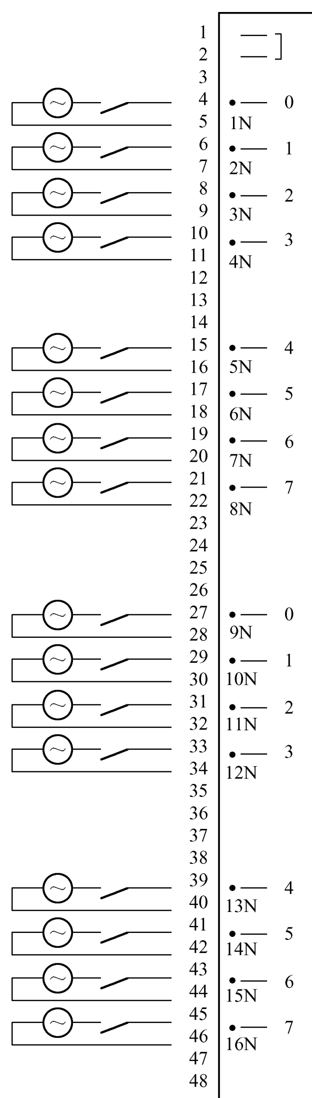



图 2-48　数字量输入模块（6ES7 421-5EH00 - 0AA0）内部结构和连接

可)，可连接普通开关（如按钮）和两线接近开关。

## 2.9.2　数字量输出模块 SM422 及其连接

数字量输出模块将二进制过程信号连接到 S7 - 400，通过这些模板，能将执行器连接到 S7 - 400，一般常见的是电磁阀、接触器、指示灯和变频器等。从输出负载的种类分，数字量输出模块主要有三种，即晶体管型（直流输出)、晶闸管型（交流输出）和继电器型（交直流输出)。目前数字量输出模块 SM422 有 8 种规格。以下将介绍有代表性的几种。

数字量输出模块（6ES7 422-1BH11-0AA0）的内部结构和连接如图 2-50 所示，SM422 模块是晶体管型，共 16 点，分两组，每组 8 点，额定电压为 DC 24 V。调试模块时，不需要给各组的 8 个输出提供负载电压（例如，1L + 和 3L + )，即使仅用 L + 为单个组供电，模块也能完全正常工作。但正常工作时，要将 1L + 、2L + 和 3L + 并联在 DC　+ 24 V 上，1 M、

2 M和 3 M 并联在 DC 0 V 上。

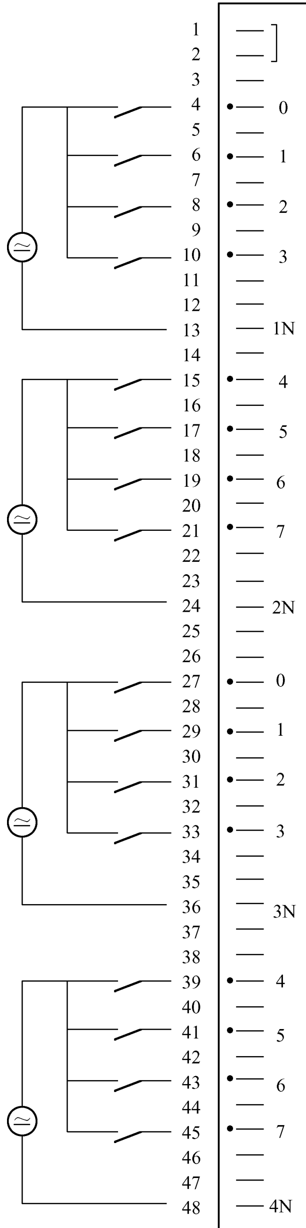



图 2-49　数字量输入模块（6ES7 421-1FH00 -0AA0）内部结构和连接

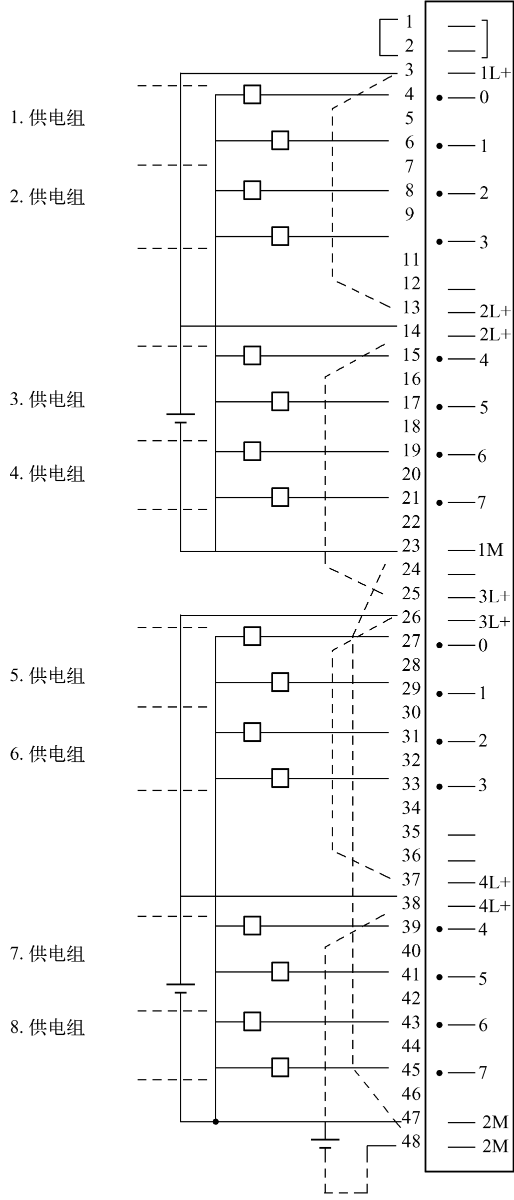



图 2-50　数字量输出模块（6ES7 422-1BH11-0AA0）内部结构和连接

数字量输出模块（6ES7 422-1FF00 -0AA0）的内部结构和连接如图 2-51 所示，该 SM422 模块是晶闸管型，共 8 点，分八组，每组 1 点，额定电压为 AC 120 V/230 V。

数字量输出模块（6ES7 422-1HH00 -0AA0）的内部结构和连接如图 2-52 所示，该 SM422 模块为继电器型，共 16 点，分八组，每组 2 点，输出电流为 5 A，额定负载电压为 AC 230 V/125 V。由于是继电器输出，所以电源的接入无极性之分，不像晶体管输出那样有极性之分。

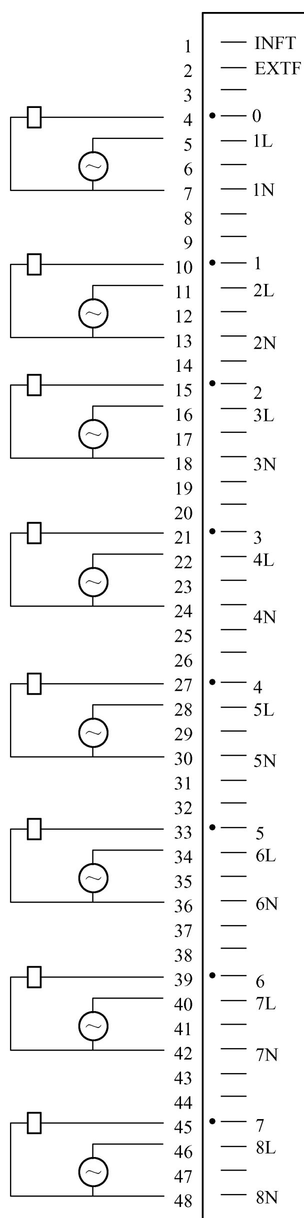



图 2-51　数字量输出模块（6ES7 422-1FF00 -0AA0）内部结构和连接

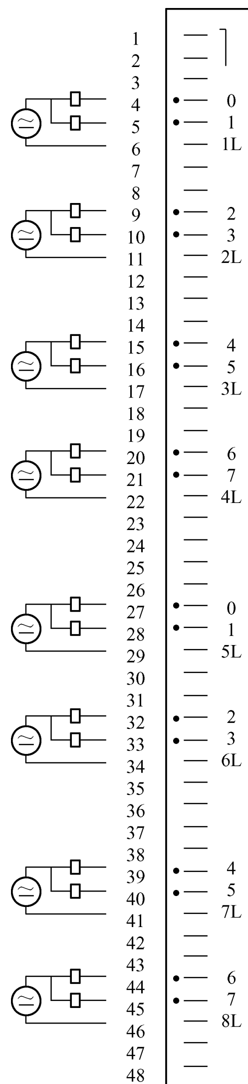



图 2-52　数字量输出模块（6ES7 422-1HH00 -0AA0）内部结构和连接

## 2.10　S7 -400 模拟量模块及其连接

S7 -400 的数字量模块包含数字量输入模块 SM431 和数字量输出模块 SM432，没有混合模块（S7 -300 有混合模块）。

### 2.10.1　模拟量输入模块 SM431 及其连接

模拟量输入模块 SM431 用于 S7 -400 模拟量输入，用于连接电压、电流、热电偶、电

阻器和热电阻，分辨率从 13～16 位。目前模拟量输入模块 SM431 有 7 种规格。以下将介绍有代表性的一种。

模拟量输入模块 SM431（6ES7431-1KF00-0AB0）简称 AI 8 x 13 位，它有 8 个输入点用于电压/电流测量，4 个输入点用于电阻测量，可并行调整的各种测量范围，准确度 13 位，模拟部分与 CPU 隔离，通道之间以及连接传感器的参考电位与 MANA 之间允许的最大共模电压为 AC 30 V。SM431 的连接如图 2-53 所示。若第一、二通道都用于测量电压信号、那么端子 6 和 9 以及端子 11 和 14 分别与 2 路电压信号连接；若第一、二通道都用于测量电流信号、那么端子 6 和 7 短接，端子 11 和 12 短接，端子 8 与 9 以及端子 13 与 14 分别与两路电流信号连接；若第一、二通道都用于测量电阻，那么只能测量一路信号，先将热电阻的一端的两个引线分别连接端子 6 和 11 上，再将热电阻的另一端的引线连到端子 9 和 14 上。

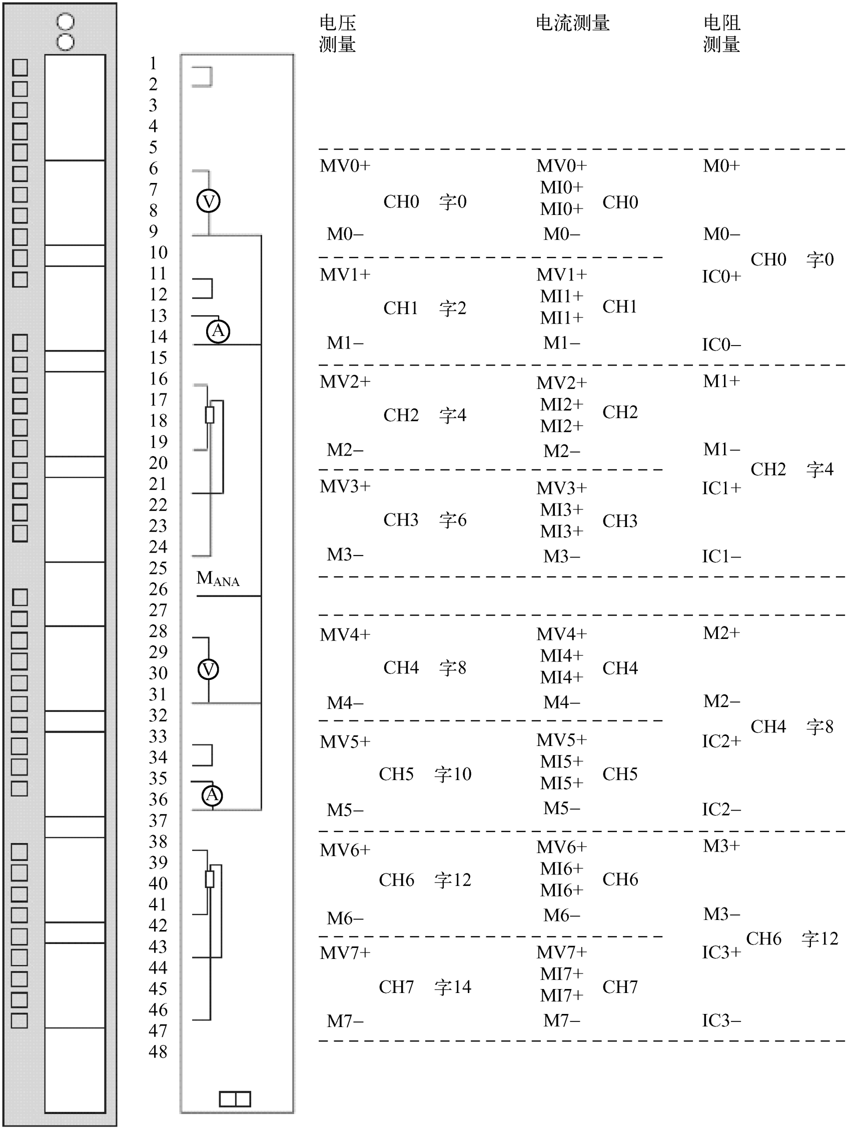



图 2-53　模拟量输入模块（6ES7 431-1KF00-0AB0）的连接

## 2.10.2 模拟量输出模块 SM432 及其连接

模拟量输出模块 SM432 用于 S7－400 模拟量输出，连接模拟量执行器。目前模拟量输出模块 SM432 有 1 个规格。

模拟量输出模块 SM432（6ES7 432-1HF00－0AB0），简称 AO8 x 13 位，有 8 点输出点，其输出通道可编程为电压输出或者电流输出（有关设置方法在后面的章节中会讲解），输出准确度 13 位，模拟量部分与 CPU 和负载电压隔离，通道之间或通道与 MANA 之间允许的最大共模电压为 DC 3 V。SM432 的连接如图 2-54 所示。

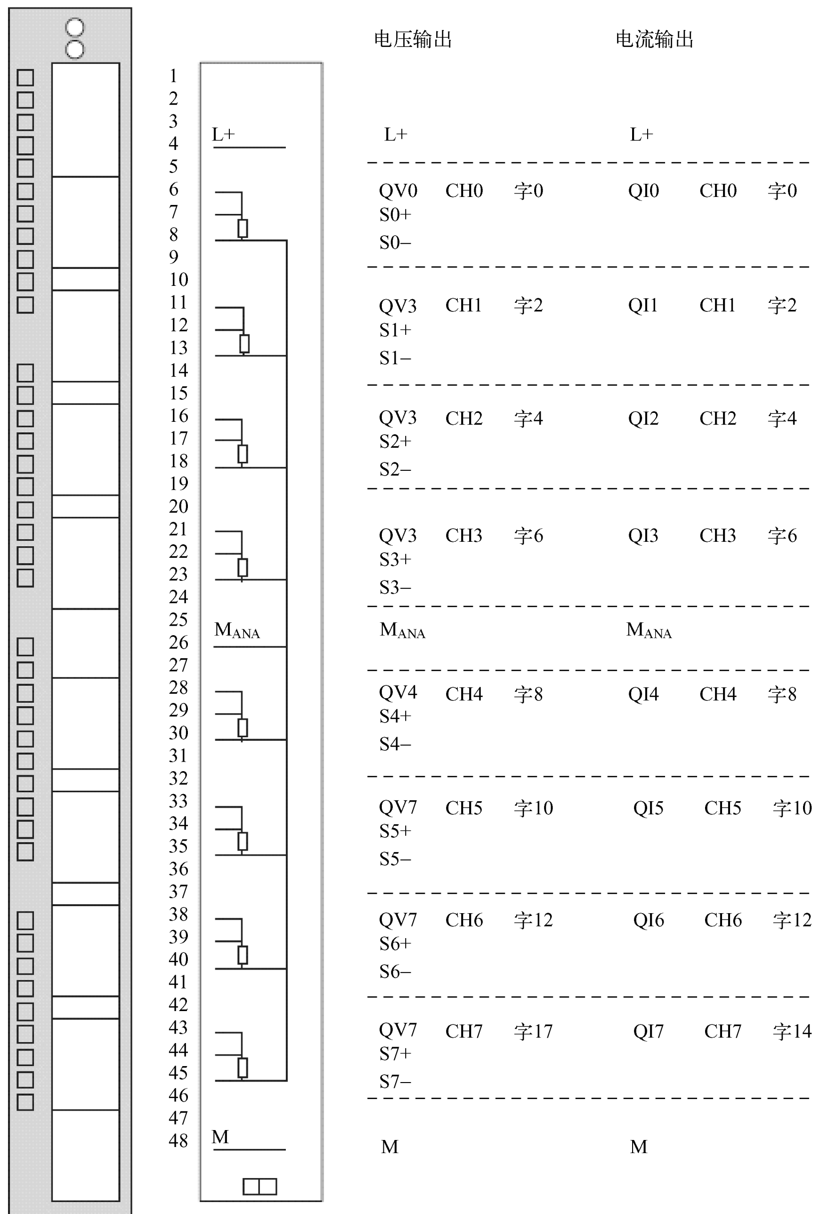



图 2-54 模拟量输出模块（6ES7 432-1HF00－0AB0）的连接

## 2.11 S7－400 的其他模块

### 2.11.1 电源模块

S7－400 的电源有 PS405 和 PS407 共两大类。电源模块用于对 S7－400 的供电，具体作用如下：

1）将 AC 或 DC 网络电压转换为所需的 DC 5 V 和 DC 24 V 工作电压，输出电流为 4 A、10 A 和 20 A 等。

2）通过 IM 向扩展机架供电。

电源的位置安装在机架的最左边，也就说从第一槽开始安装，根据配置不同，可占用 1～3 槽。

PS405 和 PS407 是有区别的，主要是接入输入电压的种类不同，PS405 电源模块输入电压为 DC 19.2～72 V，并在输出侧提供 DC 5 V 和 DC 24 V 电压。而 PS407 电源模块输入电压为 AC 85～264 V 或 DC 88～300 V，并在输出侧提供 DC 5 V 和 DC 24 V 电压。

### 2.11.2 通信处理模块

S7－400 常用的通信处理模块有：

① CP 440 通信处理器。用于利用 RS－422/RS－485（X.27）进行的短报文帧的高性能传输场合。

② CP 441-1，CP 441-2。通过点对点链接进行高速大容量串行数据交换。

③ CP 443-5。用于 PROFIBUS 通信。

④ CP 443-1 和 CP 443-1 IT。用于将 SIMATIC S7－400 连接到以太网上。

⑤ CP 444。应用 MMS 服务，根据 MAP3.0，连接工业以太网。用于减轻 CPU 的通信任务，实现深层的连接。

### 2.11.3 功能模块

S7－400 常用的功能模块有计数模块 FM450－1、定位模块 FM451 和 FM453、电子凸轮控制器 FM452 和闭环模块 FM455 等。

### 2.11.4 接口模块

S7－400 的接口模块分为两种：发送和接收模块。常用的发送模块有 IM460－0、IM460－1、IM460－2、IM460－3 和 IM460－4。常用的接收模块有 IM461-0、IM461-1、IM461-3 和 IM461-4。

# 第 3 章　STEP 7 软件使用入门

本章介绍 S7 - 300/400 PLC 编译软件 STEP 7 的使用方法，并介绍使用 STEP 7 编译一个简单程序完整过程的例子，这是学习本书后续内容的必要准备。

## 3.1　STEP 7 简介

### 3.1.1　初识 STEP 7

STEP 7 是一种用于对 SIMATIC PLC 进行组态和编程的标准软件包。它是 SIMATIC 工业软件的一部分。STEP 7 标准软件包有下列各种版本：

1）STEP 7 Micro/DOS 和 STEP 7 Micro/Win，用于 SIMATIC S7 - 200 上的简化版单机应用程序。

2）STEP 7，用于 SIMATIC S7 - 300/S7 - 400、SIMATIC M7 - 300/M7 - 400 以及 SIMATIC C7，标准 STEP 7 软件包提供一系列应用程序，具体如下：

① SIMATIC 管理器。SIMATIC 管理器（SIMATIC Manager）可以集成管理一个自动化项目的所有数据，分布式地读/写各个项目的用户数据。其他的工具都可以在 SIMATIC 管理器中起动。

② 符号编辑器。符号编辑器可以管理所有的共享符号。

③ 诊断硬件。诊断硬件的功能可以提供 PLC 的状态概况。其中可以显示符号，指示每个模块是否正常。

④ 程序编程器。用于 S7 - 300/400 的编程语言梯形图（Ladder Logic）、语句表（Statement List）和功能块图（Function Block Diagram）都集成在一个标准的软件包中。此外还有四种语言作为可选软件包使用，分别是 S7 SCL（结构化控制）编程语言、S7 Graph（顺序控制）编程语言、S7 HiGraph（状态图）编程语言和 S7 CFC（连续功能图）编程语言。

⑤ 硬件组态。硬件组态工具可以为自动化项目的硬件进行组态和参数配置。可以对机架上的硬件进行配置，设置其参数及属性。

⑥ 网络组态。网络组态（NetPro）工具用于组态通信网络连线，包括网络连接参数设置和网络中各个通信设备的参数设定，选择系统集成的通信或功能块，可以轻松实现数据的传送。

本书使用的软件是 STEP 7 V5.5CN SP4。STEP 7 V5.5CN SP4 的大部分界面已经汉化，非常适合对外语不熟悉的人使用。STEP 7 具有使用简单、面向对象、直观的用户界面、组态取代编程、统一的数据库、超强的功能、编程语言符合 IEC 1131-3 以及基于 Windows 的操作系统。

使用 STEP 7 的基本软硬件条件是一台西门子编程设备或者一台个人计算机、STEP 7 软件包和相应的许可证密匙，一台 S7 - 300/400 PLC 或者 S7 - PLCSIM。

### 3.1.2 安装 STEP 7 的软硬件条件

**1. 安装 STEP 7 V5.5 SP4 的软件系统条件**

操作系统为 Microsoft Windows XP SP2（含）以上的专业版本或者企业版、Windows Server 2003 SP2、Windows 7 专业版和旗舰（含 32 和 64 位系统）及 Windows 2008 R2（64 位系统）。

STEP 7 V5.5 SP4 与家庭版操作系统不兼容，也就是不能安装在家庭版的操作系统中。

**2. 安装 STEP 7 V5.5 SP4 的硬件系统条件**

1）奔腾处理器（600 MHz）。

2）至少 512 MB RAM。

3）彩色监视器、键盘和鼠标，Microsoft Windows 支持所有的组件。

4）硬盘空间：安装时，软件自动获取所需硬盘空间的信息。本软件的容量为 1.3 GB。

以上是安装的最低硬件要求，作者建议：CPU 的档次不低于 CORE i3，内存不低于 2 GB，否则软件的运行速度可能较慢。

### 3.1.3 安装 STEP 7 注意事项

安装 STEP 7 注意事项如下：

1）Windows 7 与 STEP 7 5.4（含）以前的版本不兼容，因此建议安装 STEP 7 5.5 SP1（或以上的版本），STEP 7 5.4（含）以前的版本，西门子公司不提供升级服务。

2）如果读者的系统是 32 位 Windows 7，那么安装仿真软件必须是 PLCSIM V5.4 SP4 及以上版本（目前暂时没有 PLCSIM V5.5 版本）。

3）如果读者的系统是 64 位 Windows 7，那么安装仿真软件必须是 PLCSIM V5.4 SP5 及以上版本。

4）西门子的大型软件对操作系统要求较高。有的盗版操作系统不能安装西门子的大型软件，即使有时能安装，有些功能也不能使用，因此要使用正版操作系统。

5）无论是 Windows 7 还是 Windows XP 系统的家庭（HOME）版都不能安装西门子的大型软件（S7-200 的除外）。

6）最好关闭监控和杀毒软件。如果电脑中安装了金山杀毒软件或者百度监控软件，则有可能不能安装仿真软件 PLCSIM，读者应先卸载金山杀毒软件或者百度监控软件，再安装仿真软件 PLCSIM，最后可安装金山杀毒软件或者百度监控软件。

7）安装软件时，软件的存放目录中不能有汉字，此时可弹出“SSF 文件错误”的信息，表明目录中有不能识别的字符。例如将软件存放在“C：/软件/STEP 7”目录中就不能安装。

8）西门子大型软件的安装顺序推荐是：首先安装 STEP 7，再安装 WinCC，最后安装 WinCC Flexible。不按照此顺序安装则可能出错。

### 3.1.4 安装 STEP 7 的过程

1）打开软件包，双击“setup.exe”，弹出如图 3-1 所示的界面，有两种安装语言供选择，本例选择“中文简体”语言，单击“下一步”按钮。

2）勾选“我接受上述许可证协议及开放源代码许可证协议的条件（A）”，单击“下一步”按钮，如图 3-2 所示。

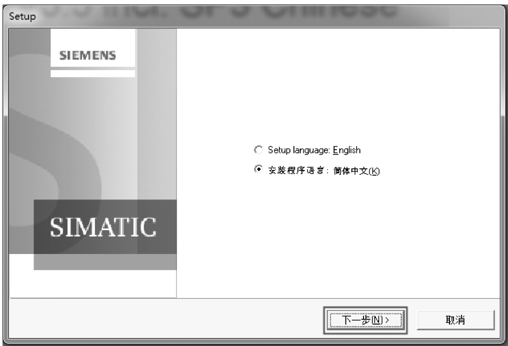



图 3-1　选择安装语言

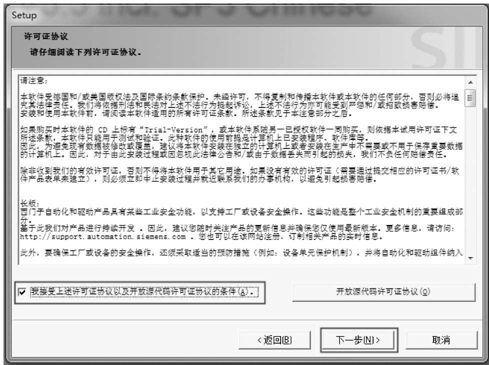



图 3-2　接受许可证协议

3）选择要安装的选项。将三个选项全部勾选，单击“下一步”按钮，如图 3-3 所示。

4）是否接受系统设置更改。勾选“接受对系统设置更改”，单击“下一步”按钮，如图 3-4 所示。

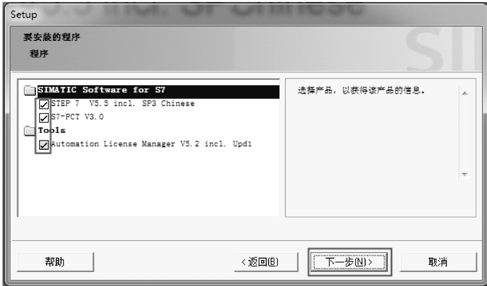



图 3-3　选择要安装的选项




图 3-4　是否接受系统设置更改

5）安装软件开始进行，如图 3-5 所示，在以后的安装过程中，仅需要单击“下一步”按钮即可。此软件的安装过程是：先安装 STEP 7 所需要的环境和插件，再安装 STEP 7 软件，最后安装小工具和帮助。

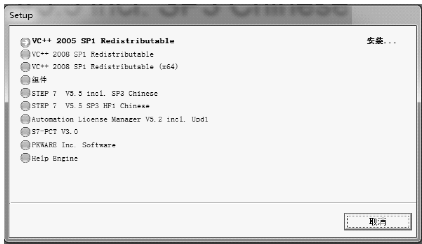



图 3-5　安装软件进行中

6）完成安装。当所有的软件安装完成后，所有的选项前面有“勾号”，如图 3-6、图 3-7 所示的安装完成的界面，一般选择“是，立即重启计算机”选项，单击“完成”按钮，之后重启计算机，再安装了授权后就可以使用 STEP 7 了。




图 3-6　完成安装（1）




图 3-7　完成安装（2）

### 3.1.5　卸载 STEP 7 的过程

使用标准 Windows 方法，卸载 STEP 7，具体过程如下：

1）在“控制面板”中双击“添加/删除程序”图标，起动 Windows 软件安装对话框。

2）在已安装软件的显示列表中选择 STEP 7 条目，单击“添加/删除”按钮。

3）出现“删除共享文件”对话框时，如果不确定，则请单击“否”按钮。

用使用标准 Windows 方法，卸载 STEP 7 有时并不彻底，有时会残留一些文件，读者可以借助如 360 等工具彻底删除残留文件。

## 3.2　编程界面的 SIMATIC 管理器

在 STEP 7 中，用项目来管理一个自动化系统的硬件和软件。STEP 7 用 SIMATIC 管理器对项目进行集中管理，它可以方便地浏览 SIMATIC S7、C7 和 WinAC 的数据。因此，掌握项目创建的方法就非常重要。

### 3.2.1　创建项目

**1. 使用向导创建项目**

首先双击桌面上的 STEP 7 图标，进入 SIMATIC Manager 窗口，进入主菜单“文件”，选择“新建项目”向导，弹出标题为“STEP 7 向导：新项目”的小窗口，如图 3-8 所示。

单击“下一步”按钮，在新项目中选择 CPU 模块的型号为 CPU 314C－2DP，如图 3-9 所示。




图 3-8　新建项目（1）




图 3-9　新建项目（2）

单击“下一步”按钮，选择需要生成的逻辑块，至少需要生成作为主程序的组织块 OB1 和编程语言 LAD（梯形图），如图 3-10 所示。

单击“下一步”按钮，输入项目的名称，单击“完成”按钮，生成的项目如图 3-11 所示。也可以单击工具栏上的“新建”按钮，新建项目。



图 3-10　新建项目（3）



图 3-11　新建项目（4）

**2. 直接创建项目**

进入主菜单“文件”，选择“新建”按钮，将出现如图 3-12 所示的一个对话框，在该对话框中分别输入“文件名称”和“存储位置”（路径）等内容，最后单击“确定”按钮，完成一个空项目的创建，如图 3-13 所示，这个项目中没有硬件、块等内容，需要组态硬件。



图 3-12　新建项目（5）

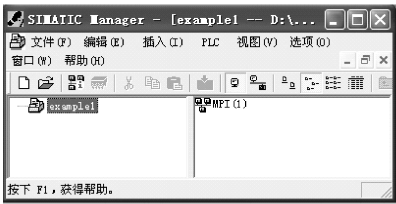


图 3-13　新建的项目（6）

## 3.2.2　编辑项目

**1. 打开已有的项目**

要打开已有的项目，选择菜单栏的“文件”→“打开”，然后选择一个项目，单击“确定”按钮，打开已有的文件。例如要打开以上创建的“example”，界面如图 3-14、图 3-15 所示。

也可以单击工具栏上的按钮，打开已有的项目。

## 2. 复制项目

复制项目的步骤是，选中要复制的项目，再在 SIMATIC 管理器中选择菜单命令“文件”→“另存为”，然后给出一个新名称，单击“确定”按钮。例如要复制以上创建的“example”，界面如图 3-16、图 3-17 所示。




图 3-14　打开项目（1）




图 3-15　打开项目（2）




图 3-16　复制项目（1）




图 3-17　复制项目（2）

## 3. 删除项目

删除项目的步骤是，在 SIMATIC 管理器中选择菜单命令“文件”→“删除”，然后选择要删除的项目，单击“确定”按钮。删除后的文件不能在回收站中找到。例如要删除以上创建的“example2”，界面如图 3-18、图 3-19 所示。




图 3-18　删除项目（1）

若要删除项目的一部分也比较容易，只要选中要删除的部分，单击鼠标右键，再单击“删除”按钮即可，例如要删除项目中的“块”，界面如图 3-20 所示。也可以选中要删除的部分，按键盘上的“删除”按钮。




图 3-19　删除项目（2）




图 3-20　删除部分项目

## 3.3　硬件组态与参数设置

### 3.3.1　硬件组态

**1. 硬件组态的任务和步骤**

在 PLC 控制系统设计前期，应根据控制系统的要求以及系统的输入/输出信号的性质和点数确定 PLC 的硬件配置，如 CPU 模块与扩展模块的型号和数量等，是否需要通信处理器模块以及种类和型号，是否需要扩展机架等。在完成上述工作后，还需要在 STEP 7 中完成硬件配置工作。

硬件组态的主要工作是根据实际系统的硬件配置，在 STEP 7 中模拟真实的 PLC 硬件配置，将电源模块、CPU 模块、信号模块及通信模块等设备安装到模拟生成的相应的机架上，生成一个与真实系统完全相同的系统，并对每个硬件组成模块进行参数设置和修改的过程。S7－300/400 的模块在出厂时已经设置默认的参数作为模块的运行参数，一般情况下，用户可以不对参数进行重新设置，这样加快了硬件组态过程。当用户确实需要修改模块的参数，需要设置网络通信等工作时，都需要硬件组态。

硬件组态的基本步骤是：生成站点→生成机架并在机架上放置模块→设置模块参数→保存参数并将它下载到 PLC 中。

**2. 硬件组态举例**

以下创建一个项目“example2”，并对此项目进行组态。

1）打开软件 STEP 7。双击“SIMATIC Manager”图标，打开 STEP 7 软件。

2）新建项目。单击工具栏上的“新建”按钮，弹出如图 3-21 所示界面，在“名称”中输入“example2”，单击“确定”按钮。

3）插入站点。选中管理器中的项目名称“example2”，再单击菜单栏中的插入→站点→SIMATIC 300 站点，如图 3-22 所示。

4）起动硬件组态界面。单击项目名称“example2”左侧的“＋”，选定 SIMATIC300（1）→双击硬件，之后弹出硬件组态界面，如图 3-23 所示。

5）插入机架。在硬件组态界面中，双击机架“Rail”，机架自动弹出，如图 3-24 所示。



图 3-21　新建项目

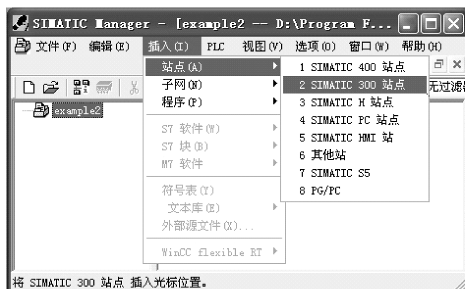


图 3-22　插入站点



图 3-23　起动硬件组态界面

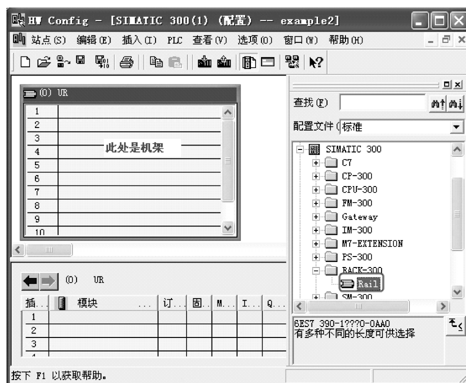


图 3-24　插入机架

6）插入电源模块。在硬件组态界面中，先选中“1”处，再双击“PS 307 5 A”，电源模块自动安装到 1 号槽位，如图 3-25 所示。也可以用鼠标的左键选中“PS 307 5 A”，并按住不放，将电源模块拖至 1 号槽位。

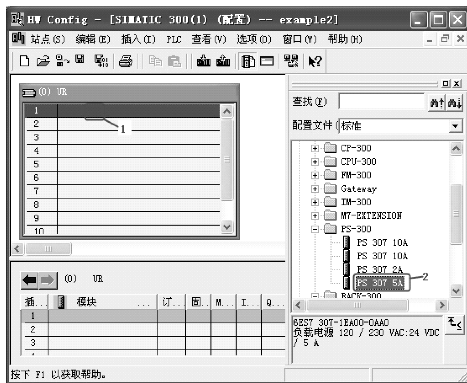


图 3-25　插入电源模块

7）插入 CPU 模块。在硬件组态界面中，用鼠标的左键选中“V2.6”，并按住不放，将 CPU 模块拖至 2 号槽位，如图 3-26 所示。也可以先选中“1”处，再双击“CPU 314 - 2 DP”下的“V2.6”，CPU 模块自动安装到 2 号槽位。

8）保存和编译组态。至此一个简单的项目已经组态完成，只要单击工具栏上的“保存和编译”按钮即可，如图 3-27 所示。

## 3.3.2　参数设定

利用 STEP 7，除了可以完成西门子 S7 - 300/400 系列 PLC 的硬件组态外，还可以用来设置模块的参数。

**1. CPU 参数的设定**

打开 HW Config 硬件组态界面，双击 CPU 模块所在的行，弹出属性窗口，选择某一选项卡，便可对其相应的属性进行设置。

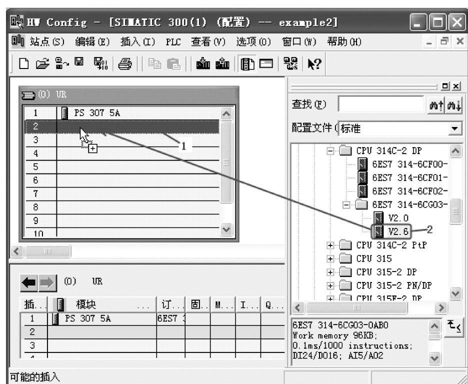


图 3-26　插入 CPU 模块（用拖放方法）

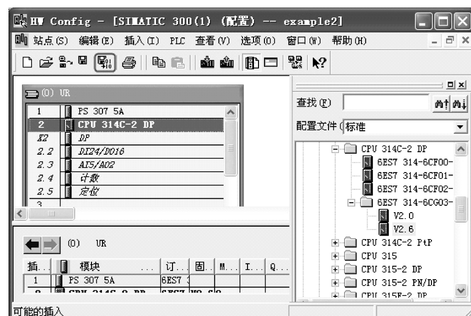


图 3-27　保存和编译组态

（1）CPU 的起动参数设置

CPU 的起动性能参数可以在“属性”窗口中的“启动”选项卡中设置，如图 3-28 所示。

“启动”选项卡中有“如果预先设置的组态与实际组态不相符合则启动”选项，若选中这个选项，复选框的方框中应该有“√”，当模块没有插在组态时指定的槽位或者某个槽位实际插入的模块与组态的模块不相符合时，CPU 仍然会启动；若没有选中这个选项，则当出现上述情况时，CPU 将进入 STOP 状态。

“热启动时复位输出”和“禁止操作员热启动”选项仅用于 S7 -400 CPU，S7 -300 CPU 中该选项是灰色的。

“通电后启动”是用于设置电源接通后的启动选项，可以选择单选按钮“热启动”“暖启动”和“冷启动”。CPU -318 和 CPU417 -4 具有冷启动方式，冷启动时，所有的过程映像区和标志存储器、定时器和计数器（无论是保持型还是非保持型）都将被清零，而且数据块的当前值被装载存储器的原始值覆盖。暖启动方式起动时，过程映像区和不保持的标志存储器、定时器及计数器被清零，保持的标志存储器、定时器和计数器以及数据块的当前值保持原状态。一般 S7 -300 PLC 都采用此种启动方式。热启动方式启动时，所有数据（无论是保持型和非保持型）都将保持原状态。然后程序从断点处开始执行。热启动一般只有 S7 -400具有此功能。

“监视时间”选项用于设置相关项目的监控时间。

“来自模块的完成消息”选项用于设置电源接通后，CPU 等待所有被组态的模块发出“完成信息”的时间。“参数传递到模块的时间”用于将参数传递到模块的最长时间。

（2）扫描周期/时钟存储器的参数设置

扫描周期/时钟存储器的参数设置通过“周期/时钟存储器”选项卡来设置，如图 3-29 所示。

“扫描周期监视时间”选项用于设置循环扫描时间，以“毫秒”为单位，默认值为 150 毫秒，当实际扫描时间大于设定值时，CPU 进入 STOP 模式。

时钟存储器用于设置时钟存储器的字节地址。S7 -300/400 提供了一些不同频率的、占

空比为1∶1的方波脉冲信号给用户程序使用，这些方波信号存储在一个字节的时钟存储器中，该字节默认为MB0（此地址允许修改），需要将前面的复选框选中才能激活，该字节的每一位对应一种频率时钟脉冲信号，见表3-1。

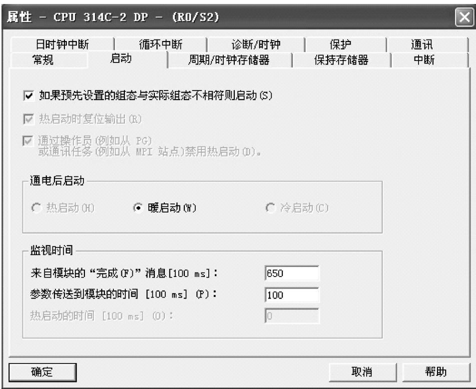



图3-28　CPU的属性窗口-启动参数设置

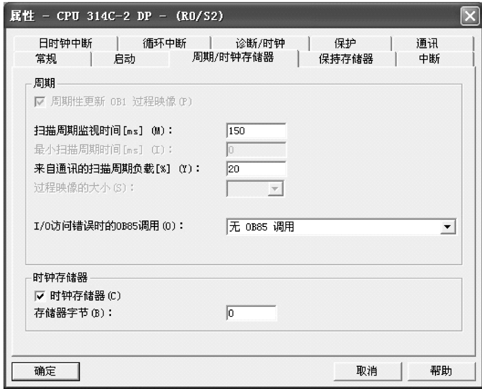



图3-29　CPU的属性窗口-周期/时钟存储器参数设置

**表3-1　时钟存储器位与时钟脉冲周期和频率对应表**

| 位 | 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 |
|---|---|---|---|---|---|---|---|---|
| 周期/s | 2 | 1.6 | 1 | 0.8 | 0.5 | 0.4 | 0.2 | 0.1 |
| 频率/Hz | 0.5 | 0.625 | 1 | 1.25 | 2 | 2.5 | 5 | 10 |

例如，图3-29中的时钟存储器为MB0，所以M0.5的频率为1 Hz，可以用在明暗闪烁频率为1 Hz的报警灯中。

（3）系统诊断与时钟的参数设置

系统诊断与时钟的参数设置可以通过属性窗口的“诊断/时钟”选项卡设置，如图3-30所示。

通过系统诊断可以发现用于程序的错误、模块的故障以及传感器和执行器的故障等。故障诊断可以通过选择“报告STOP模式原因”等选项设置。




图3-30　CPU的属性窗口-诊断/时钟参数设置

（4）保持存储器的参数设置

保持存储器的参数设置可以通过属性窗口的“保持存储器”选项卡设置，如图3-24所示。

所谓保持存储器就是电源掉电或CPU从RUN进入STOP模式后内容保持不变的存储区。安装了后备电池的S7系列PLC，用户程序中的数据块总是保存在保持存储区，没有后备电池的PLC可以在数据块中设置保持区域。

图3-31中默认的保持存储区为MB0～MB15、C0～C7。

（5）口令保护和运行方式的设置

口令保护和运行方式的设置可以通过属性窗口的“保护”选项卡设置，如图3-32所示。

在 S7－300/400 系列 PLC 中，使用口令保护功能可以对 CPU 中的程序和数据进行保护，有效防止控制过程中可能的人为干扰。设置完成后将其下载到 CPU 模块中。

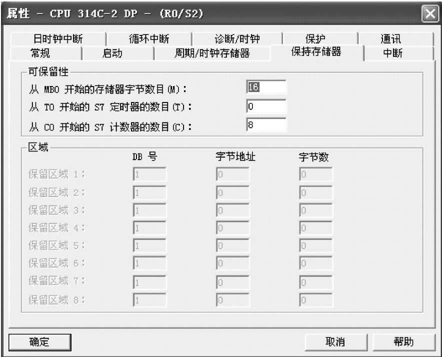



图 3-31　CPU 的属性
窗口－保持存储器参数设置

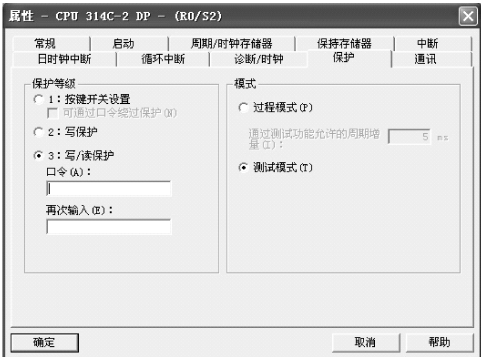



图 3-32　CPU 的属性
窗口－口令保护和运行方式的设置

“保护”选项卡中有三个保护级别：保护级别 1，不需要设置口令；保护级别 2，只能进行读操作，而不能进行写操作；保护级别 3，只有拥有授权，才能进行读/写操作。口令的设置很容易，在图 3-32 的“口令”的方框中输入字母或者数字，单击“确定”按钮即可。

（6）中断参数的设置

如图 3-33 所示的“中断”选项卡中，可以设置硬件中断、时间延迟中断和异步错误中断的中断优先级。默认情况下，所有的硬件中断都由 OB40 来处理，用户可以通过设置优先级屏蔽中断。

（7）循环中断参数的设置

如图 3-34 所示的“循环中断”选项卡中，可以设置循环执行组织块 OB30～OB38 的参数，这些参数包括中断优先级、以毫秒为单位的执行时间间隔和相位偏移量。例如，图 3-34 中每 100 ms 执行组织块 OB35 中的程序。

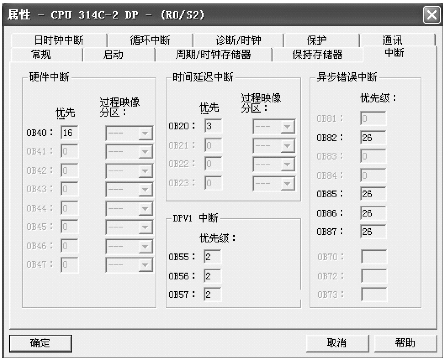



图 3-33　CPU 的属性
窗口－中断参数的设置

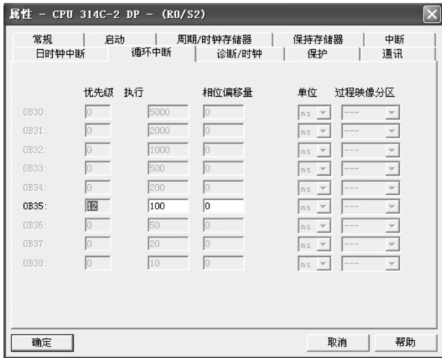



图 3-34　CPU 的属性
窗口－循环中断参数的设置

**2. 数字量输入/输出模块的参数设置**

数字量输入/输出模块的参数分为动态参数和静态参数，在 CPU 处于 STOP 模式时，通过 STEP 7 的硬件组态，可以设置两种参数，参数设置完成后，应将参数下载到 CPU 中，这样当 CPU 从 STOP 转为 RUN 模式时，CPU 将参数自动传送到每个模块。

用户程序运行时，可以通过系统 SFC 调用修改动态参数。但当 CPU 从 RUN 模式进入 STOP 又返回 RUN 模式后，PLC 的 CPU 将重新传送 STEP 7 设置的参数到模块中，动态设置的参数丢失。

（1）数字输入模块的参数设置

打开 HW Config 硬件组态界面，双击数字量输入模块所在的行，便可弹出属性窗口，选择某一选项卡，便可对其相应的属性进行设置。

在“地址”选项卡中可以设置数字量输入模块的起始字节的地址。若要修改起始地址，先要把“系统默认”前的复选框上的“√”去掉，再在“开始”后的框中输入新的起始地址即可，最后单击“确定”按钮，如图 3–35 所示。

对于有中断功能的数字量输入模块，还有“输入”选项卡。可以通过复选框，选择是否允许产生“诊断中断”和“硬件中断”。如图 3–36 所示，先激活“硬件中断”，再激活输入点 0 和 1 位的“上升沿”，最后单击“确定”按钮，这样设置的含义是当这个数字量输入模块的第 0 或者第 1 位有上升沿时，触发硬件中断，CPU 将调用 OB40 进行处理。



图 3–35　数字输入量模块 – 地址参数的设置



图 3–36　数字输入量模块 – 输入参数的设置

（2）数字输出模块的参数设置

在“地址”选项卡中可以设置数字量输出模块的输出起始地址，设置方法和数字量输入模块的类似。注意，当重新设置的地址若已经占用，则修改是不能成功的。

有些输出模块有诊断中断、输出强制值功能，可以在“输出”中设置。在该选项卡中，单击复选框，可以设置是否允许产生诊断中断。“对 CPU STOP 模式的响应”下拉列表可以选择 CPU 进入 STOP 模式时，模块对各输出点的处理方式。选择“保持前一个有效值”，则 CPU 进入 STOP 模式后，模块将保持最后的输出值；而选择“替换值”，CPU 进入 STOP 模式后，可以使各点输出一个固定值，该值由“替换值 1”选项的复选框决定。如图 3–37 所示，替换值的所有复选框都激活，所以当 CPU 进入 STOP 模式后，所有选中的输出点都为“1”。

**3. 模拟量输入/输出模块的参数设置**

（1）模拟量输入模块的参数设置

模拟量输入模块的地址可以在“地址”选项卡中修改，方法与数字量模块的类似。模拟量模块“输入”选项卡如图 3–38 所示。




图 3-37　数字输出量模块 – 输出参数的设置




图 3-38　模拟输入量模块 – 输入参数的设置

如果已激活“诊断中断”复选框并发生诊断事件，则相应信息会输入到模块的诊断数据区。然后，模块会触发诊断中断，S7 – CPU 会调用诊断中断块 OB82（将在后面的章节讲解）。如果已激活“超出限制时硬件中断”复选框，输入值超出“上限值”和“下限值”定义的范围，模块会触发硬件中断。

在“输入”选项卡中，还可以对模块的每一个通道的测量类型和测量量程。单击通道组的“量程型号”输入框，在弹出的菜单中选择测量种类，其中“E”表示测量电压信号、“4DMU”表示 4 线传感器的电流信号测量、“2DMU”表示 2 线传感器的电流信号测量、“R – 4L”表示 4 导线电阻测量、“RT”表示热敏电阻测量温度信号、“TC – I”表示热电偶测量温度，如图 3-39 所示。单击“测量范围”输入框，弹出测量范围菜单（供选择，仅以测量电压信号为例说明），若输入电压信号的范围是 –5 ~ +5 V，则选择“ –5 ~ +5 V”选项，如图 3-40 所示。

模拟量模块的使用时，量程卡的设置非常重要，每个量程卡与 2 个通道关联。如图 3-41 所示的量程卡为“A”位置，这个位置是根据硬件组态结果设置的，如图 3-38 所示，第 0 和第 1 通道两个关联通道要求量程卡设置在“A”位置，因此读者要用螺钉旋具撬开量程卡，将“A”位置对准量程卡上的“三角”标识。

| | |
|---|---|
| | 取消激活 |
| E | 电压 |
| 4DMU | 电流(4 线传感器) |
| 2DMU | 电流(2 线传感器) |
| R-4L | 电阻(4 导线端子) |
| RT | 电阻(热敏，线性) |
| TC-I | 热电偶(内部补偿) |
| TC-E | 热电偶(外部补偿) |
| TC-IL | 热电偶(内部补偿线性) |
| TC-EL | 热电偶(外部补偿线性) |

图 3-39　测量类型的设置

+/- 80 mV
+/- 250 mV
+/- 500 mV
+/- 1 V
+/- 2.5 V
+/- 5 V
1..5 V
+/- 10 V

图 3-40　测量范围（电压）的设置

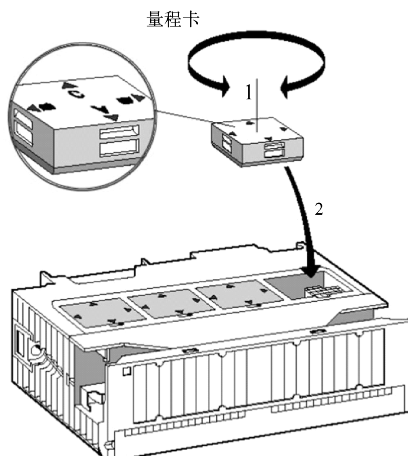



图 3-41　量程卡的设置方法

对于不使用的模拟量通道的处理方法，在前一章已有讲解，但关联在一起的两个通道都不使用，则取消激活，前述章节不容易说明清楚，在此补充说明。例如第 6 和第 7 通道两个关联通道，不使用，则做如图 3-42 所示的处理，即取消激活，使用这种方法处理不使用的通道最简单，但注意这两个通道必须是关联在一起的，如第 6 和第 7 通道就是关联在一起的，可采用这种方法处理，而第 5 和第 6 通道就没有关联在一起，处理方法见前述章节。

（2）模拟量输出模块的参数设置

模拟量输出模块的地址可以在“地址”选项卡中修改，方法与数字量模块的类似。模拟量模块“输出”选项卡如图 3-43 所示。




图 3-42　取消激活

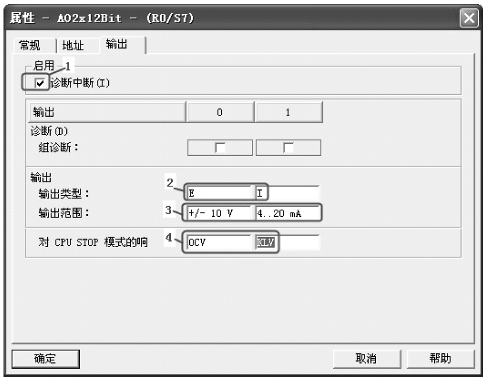



图 3-43　模拟输出量模块 - 输出参数的设置

可以设置各个通道是否允许中断，含义与模拟量输入模块的相同。

模拟量的“输出类型”有三种即“电流”“电压”和“取消”，只要单击通道“测量类型”的方框，在弹出的菜单中选取即可。“电流”的含义是输出的模拟信号是电流信号，“电压”的含义是输出的模拟信号是电压信号，“取消”的含义是不使用此通道。

“输出”范围的含义是输出模拟信号的范围，选择的方法是只要单击通道“输出范围”的方框，在弹出的菜单中选取即可。

对 CPU STOP 模式的响应有三个选项，其中“OCV”的含义是不输出电流电压，“KLV”的含义是保持前一个输出的电流电压值，“SV”的含义是采用替代值。

## 3.3.3　硬件的更新和 GSD 文件安装

### 1. 硬件的更新

西门子的硬件更新比较快，每一个 STEP 7 版本都不可能包含未来的硬件。从 STEP 7 5.2 开始，STEP 7 提供了硬件更新功能。在组态硬件时，有时无法找到需要配置的硬件（一般以订货号为准），这就需要更新目录中的硬件。

硬件的更新有两种方法，一种是从磁盘上复制更新，另一种是从互联网上更新，这种形式比较方便。方法如下：

首先打开 STEP 7 的硬件组态界面，单击菜单栏的“选项”→“安装 HW 更新”，如图 3-44 所示，弹出要求选择的更新的路径，如图 3-45 所示，选择“从 Internet 下载”更新，再单击“执行”按钮，进入自动搜索需要下载的硬件，如图 3-46 所示。




图 3-44　打开“硬件更新”的路径




图 3-45　从 Internet 上更新

搜索完毕后，弹出要求下载的硬件列表，选择需要下载的硬件，如图 3-47 所示，再单击“下载”按钮，下载完成后，自动弹出已经下载的硬件列表，如图 3-48 所示，选择需要安装的硬件并单击“安装”按钮，按照提示安装，安装时，STEP 7 要关闭。安装完成后重新打开 STEP 7，重新安装的硬件就可以使用了。




图 3-46　下载过程

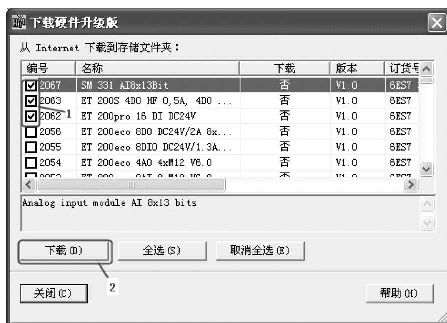



图 3-47　选择需要下载的硬件




图 3-48　选择需要安装的硬件

**2. 安装 GSD 文件**

（1）GSD 文件简介

PROFIBUS 设备具有不同的性能特点，为达到 PROFIBUS 简单的即插即用配置。PROFIBUS 设备的特性均在电子设备数据库文件（GSD）中具体说明。标准化的 GSD 数据将通信扩大到操作员控制级。使用基于 GSD 的组态工具可将不同厂商生产的设备集成在同一总线系统中，既简单又是对用户友好的。

（2）安装 GSD 文件

例如，安装了 STEP 7 V5.5 是不能组态 EM277 的（S7 -300/400 与 S7 -200 进行 PROFIBUS 通信需要组态此模块），因为 STEP 7 V5.5 中没有 SIEM089D.GSD 文件，因此必须安装此文件才能组态 EM277 模块。安装 SIEM089D.GSD 文件的方法如下：

首先打开 STEP7 的硬件组态界面，单击菜单栏的“选项”→“安装 GSD 文件”，如图 3-49所示，弹出要求选择的安装 GSD 文件的路径，如图 3-50 所示，先选择 GSD 文件存放的目录，本例为“D:\softWare\GSD”，再选择要安装的 GSD 文件，本例为“SIEM089D. GSD”，最后单击“安装”按钮即可。




图 3-49　打开“安装 GSD 文件”的路径




图 3-50　选择需要安装的硬件

## 3.4　下载和上传

### 3.4.1　下载

STEP 7 可以把用户的组态信息（SDB－系统数据）和程序下载到 CPU 中。下载的常用方法包含：在 SIMATIC 管理器画面、在具体的程序或组态画面和离线/在线画面中进行下载。

在下载过程中，STEP 7 会提示用户处理相关信息，如：是否要删除模块中的系统数据，并离线系统数据替代、OB1 已经存在是否覆盖以及是否停止 CPU 等，用户应该按照这些提示做出选择，完成下载任务。

当下载时把 CPU 面板的模式开关切换到“RUN”或“RUN－P”模式，下载信息包含硬件组态信息或网络组态信息等系统数据时，会提示需要切换到停止状态。

（1）在项目管理器中下载

在管理器中下载既可以下载整个站，也可以只下载一部分，例如只下载一个块。先介绍下载整个站的步骤。

首先打开 STEP 7 的项目管理器界面，单击菜单栏的“选项”→“设置 PG/PC 接口”，如图 3-51 所示，弹出“设置 PG/PC 接口”界面，如图 3-52 所示，选择“PC Adapter（MPI）”，单击“属性”按钮。

**【关键点】** 下载软硬件的方法很多，可以用 MPI、PROFIBUS、S7、ISO 和 TCP 等通信协议下载，可以采用 MPI 适配器、以太网和 CP5621 卡（还有 CP5511 等）传输介质下载，本例仅使用 MPI 协议，用 MPI 适配器下载程序，还有很多其他下载程序的方法。

在“MPI”选项卡中，设置编程器（或者 PC）的地址，本例为“0”，在设定传输率为“187. 5 kbps”，如图 3-53 所示。




图 3-51　打开“设置 PG/PC 接口”界面




图 3-52　“设置 PG/PC 接口”界面

**【关键点】** 编程器（或者 PC）的地址是唯一的，一般选用默认值“0”，传输率一般也选用默认值“187. 5 kbit/s”。

在“本地连接”选项卡中，设置编程器（或者 PC）的通信接口，本例为“USB”，再单击“确定”按钮即可，如图 3-54 所示。

**【关键点】** MPI 适配器一端与编程器相连，通常是 USB 接口或者 RS－232C 接口，另一端与 PLC 相连，是 RS－485 接口。若 MPI 适配器上有 USB 接口，使用的是 STEP7 V5. 4（含）以前的版本，必须在 STEP 7 中安装驱动程序，此程序可以在西门子的官方网站（http://www. ad. siemens. com. cn/）上下载。STEP 7 V5. 5（含）以后的版本无需安装驱动程序。

此外，使用 64 位操作系统没有选择“本地连接”中的“USB”这个步骤。

回到如图 3-52 所示的界面，单击“确定”按钮，弹出如图 3-55 所示界面，单击“确定”按钮。




图 3-53　通信参数设置




图 3-54　设置通信接口

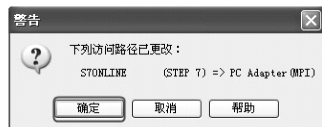



图 3-55　更改访问路径

**【关键点】** 若上一次使用的通信协议与这次相同，则不会有此界面弹出。

先选取站点“S7－300”，再单击“下载”按钮，如图 3-56 所示，程序和硬件组态开始下载，若组态模块的信息和实际目标模块的信息不同时，弹出如图 3-57 所示界面，若订货号等有报警（即出现一个黄色三角形，三角形中有一个“!”），则必须重新组态。

如果 PLC 中已经有程序或者硬件组态，则会弹出如图 3-58 所示界面，含义为是否将 CPU 中的“OB1”用编程器中的替代（或者说覆盖），单击“全部”按钮，表示全部覆盖。当所有的程序下载完成后，弹出如图 3-59 所示界面，提示是否重新起动 CPU，单击“是”

按钮，则重启 CPU。

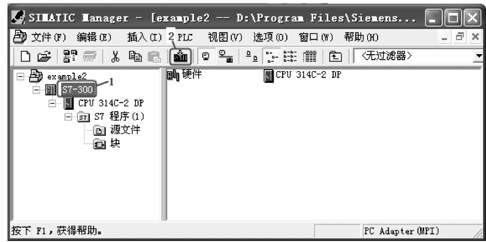


图 3-56　项目管理器画面




图 3-57　组态模块的信息和实际目标模块的信息对照

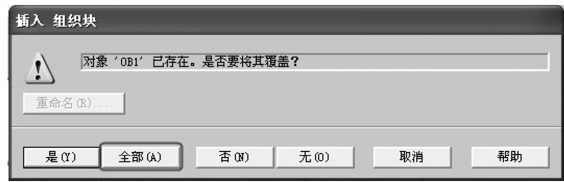



图 3-58　是否“覆盖 OB1”信息

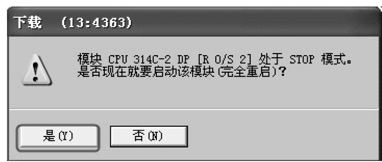



图 3-59　重启信息

（2）在具体的程序或者组态中下载

如果硬件组态已经下载到 CPU，则接下来只要下载程序即可，没有必要每次下载时，都将整个项目都全部下载。

首先打开 STEP 7 的程序编辑器界面，再单击“下载”按钮即可，如图 3-60 所示。

【关键点】程序有语法错误是不能下载的。

如果硬件组态和程序已经下载到 CPU，只对硬件组态做了修改，没有必要每次下载时，将整个项目都全部下载，只下载硬件组态即可。

首先打开 STEP 7 的硬件组态界面，再单击“下载”按钮即可，如图 3-61 所示。

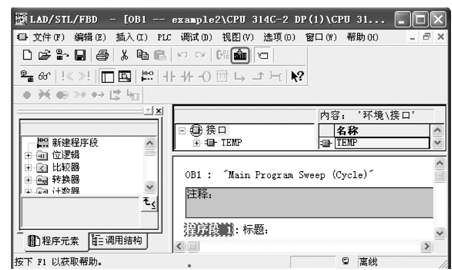


图 3-60　在“程序编辑器”中下载程序

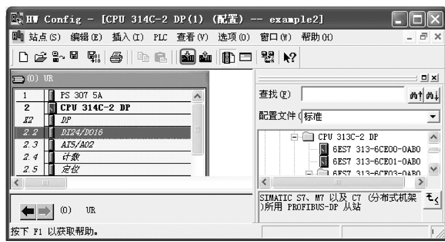


图 3-61　在“硬件组态界面”中硬件组态

【关键点】硬件组态中错误是不能下载的。

## 3.4.2　上传

STEP 7 可以把 CPU 中的组态信息和程序上传到用户的项目中。上传常用的方法有：在 SIMATIC 管理器界面、在硬件组态界面和在线/离线界面中进行上传。

**1. 在 SIMATIC 管理器中上传**

在 SIMATIC 管理中先新建一个空项目“Upload”，单击菜单栏“PLC”→“将站点上传

到 PG”，如图 3-62 所示，弹出选择节点地址对话框，先选择目标站点为“本地”，再单击“更新”按钮，如图 3-63 所示，再单击“确定”按钮，上传开始，上传过程如图 3-64 所示。




图 3-62 在项目管理中上传的路径

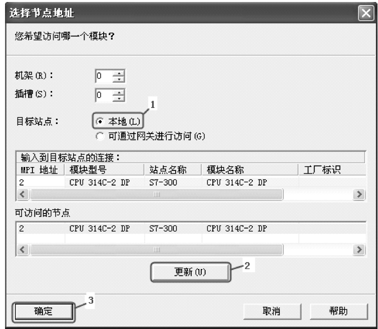



图 3-63 选择节点地址对话框

【关键点】用这种方法上传的是站点，包括硬件组态信息和程序。

**2. 在组态界面中上传**

在组态界面中上传，先打开 STEP 7 的硬件组态界面，如图 3-65 所示，单击工具栏的“上传”按钮，弹出选择目标项目对话框，先选择目标项目，本例为“Upload”，再单击“确定”按钮，弹出选择上传节点地址对话框，如图 3-66 所示，弹出选择节点地址对话框，先选择目标站点为“本地”，再单击“更新”按钮，如图 3-67 所示，再单击“确定”按钮，上传开始，上传过程如图 3-64 所示。

【关键点】用这种方法上传的信息只有硬件组态信息。

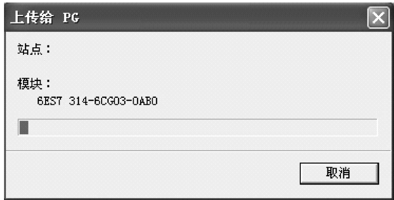



图 3-64 上传过程中




图 3-65 上传




图 3-66 选择目标项目

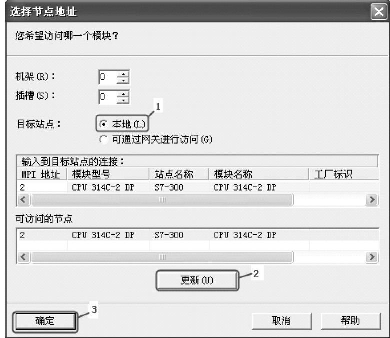



图 3-67 选择上传节点地址对话框

## 3.5 软件编程

对于任何 PLC 项目，编写程序总是必需的，编写程序在硬件组态完成后，主程序编写在 OB1 组织块中，其他程序如时间循环中断程序可编写在 OB35 中。以下介绍一个最简单的程序的输入和编译过程。

1）打开程序编译器。首先在 SIMATIC 管理器界面，选中并双击“OB1”组织块，打开程序编辑器界面，如图 3-68 所示。

2）输入程序。选中如图 3-69 所示的“1”处，双击常开触点“┤├”，常开触点自动跳到“1”处，再用同样的方法双击线圈“-()”，线圈跳到常开触点的后面，如图 3-70 所示。

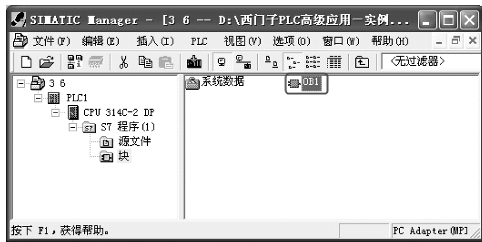



图 3-68　SIMATIC 管理器界面




图 3-69　输入程序（1）

单击常开触点“┤├”上的红色“I0.0”，弹出方框，在方框中输入“I0.0”，单击回车键即可，同理，输入“Q0.0”，按回车键，如图 3-71 所示。

3）保存程序。单击工具栏上的“保存”按钮即可。




图 3-70　输入程序（2）

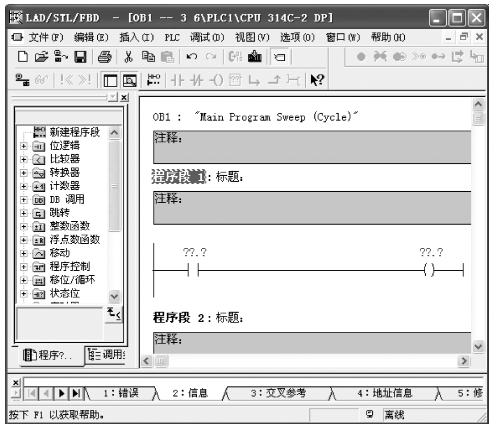



图 3-71　输入程序（3）

## 3.6 打印和归档

一个完整的项目包含文字、图表和程序文件。打印的目的就是进行纸面上的交流和存档，项目归档是电子方面的交流和存档。

### 3.6.1 打印

打印的操作步骤如下：

1）打开相应的项目对象，在屏幕上显示要打印的信息。

2）在应用程序窗口中，使用菜单栏命令“文件”→“打印”，打开打印界面。

3）可以在对话框中更改打印选项（例如打印机、打印范围和打印份数等）。

也可以将程序等生成 mdi 或者 pdf 格式的文档。以下介绍生成 mdi 格式文档的步骤。

在程序编辑器中，使用菜单栏命令“文件”→“打印”，打开打印对话框，如图 3-72 所示，打印机名称选择“Microsoft Office Document Image Writer”，再单击“确定”按钮，生成的 mdi 格式文档如图 3-73 所示。

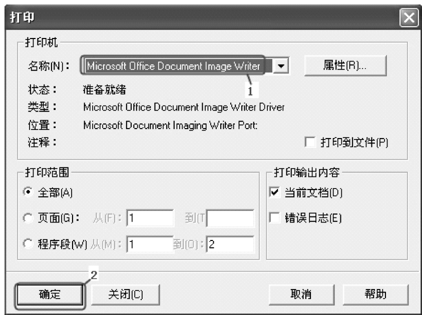



图 3-72　打印对话框

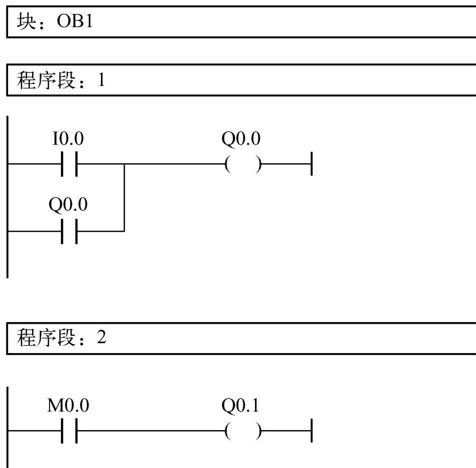



图 3-73　程序生成 mdi 格式的文档例子

### 3.6.2 归档

项目归档的目的是把整个项目的文档压缩到一个压缩文件中，以方便备份和转移。当需要使用时，使用解压的方式恢复为原来项目的文档。归档的步骤如下：

打开项目 SIMATIC 资源管理器，单击菜单栏的“文件”→“归档”，如图 3-74 所示，弹出查找将要归档的项目名称，如图 3-75 所示，单击

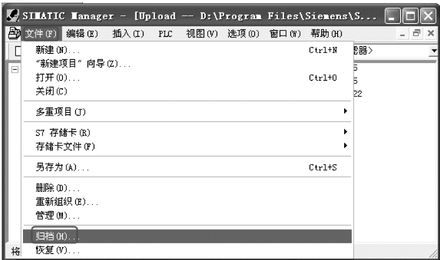



图 3-74　归档的路径

“确定”按钮，弹出选择归档的路径，如图 3-76 所示，单击“保存”按钮，生成一个后缀为“. ZIP”的压缩文件。




图 3-75　选择将要归档的项目名称




图 3-76　选择归档的路径

## 3.7　用 STEP 7 V5.5 建立一个完整的项目

以下创建一个项目，用一台 CPU314C－2DP 控制一盏灯的开和关。

1）先设计原理图，如图 3-77 所示，并按照原理图接线。

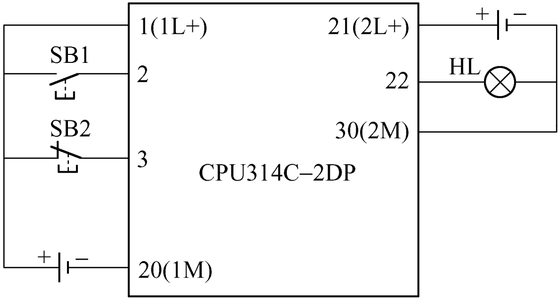



图 3-77　原理图

2）打开软件 STEP 7，新建项目，并进行硬件组态。

① 打开软件 STEP 7。双击“SIMATIC Manager”图标，打开 STEP 7 软件。

② 新建项目。单击工具栏上的“新建”按钮，弹出界面如图 3-78 所示，在“名称”中输入“example”，单击“确定”按钮。

③ 插入站点。选中 SIMATIC 管理器中的项目名称“example”，再单击菜单栏中的插入→站点→SIMATIC 300 站点，如图 3-79 所示。

④ 起动硬件组态界面。单击项目名称“example”左侧的“＋”，选定 SIMATIC 300（1）→双击硬件，之后弹出硬件组态界面，如图 3-80 所示。

⑤插入机架。在硬件组态界面中，双击机架“Rail”，机架自动弹出，如图 3-81 所示。

⑥ 插入电源模块。在硬件组态界面中，先选中“1”处，再双击“PS 307 5A”，电源模块自动安装到 1 号槽位，如图 3-82 所示。也可以用鼠标的左键选中“PS 307 5A”，并按住不放，将电源模块拖至 1 号槽位。




图 3-78　新建项目

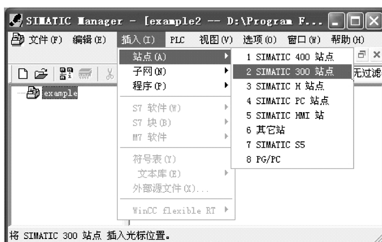



图 3-79　插入站点

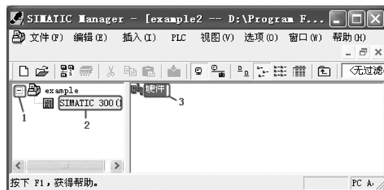



图 3-80　起动硬件组态界面

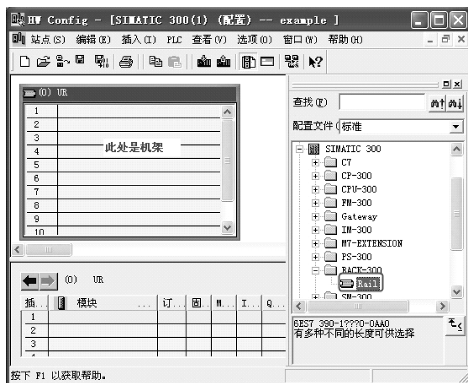



图 3-81　插入机架

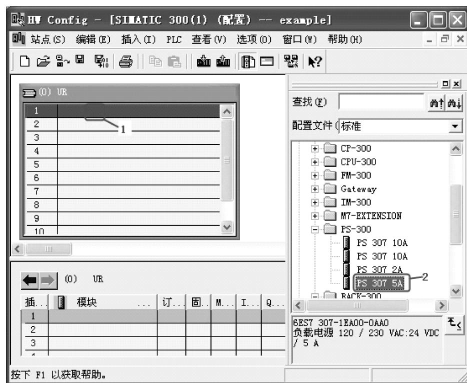



图 3-82　插入电源模块

⑦ 插入 CPU 模块。在硬件组态界面中，用鼠标的左键选中“V2.6”，并按住不放，将 CPU 模块拖至 2 号槽位，如图 3-83 所示。也可以先选中“1”处，再双击“CPU 314 - 2 DP”下的“V2.6”，CPU 模块自动安装到 2 号槽位。

⑧ 保存和编译组态。至此一个简单的项目已经组态完成，只要单击工具栏上的“保存和编译”按钮即可，如图 3-84 所示。

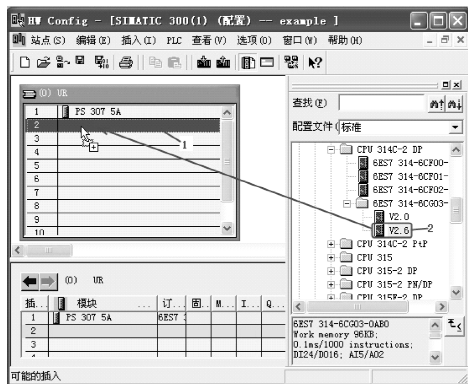



图 3-83　插入 CPU 模块（用拖放方法）




图 3-84　保存和编译组态

3）编写程序。打开程序编辑器，在编辑器中编写程序，如图 3-85 所示，再单击工具栏上的“保存”按钮，保存程序。

【关键点】CPU314C－2DP 的集成数字 I/O 的默认起始地址是 IB124 和 QB124，当然这个地址是可以修改的，如改成 IB0 和 QB0。

4）下载硬件组态和程序。先把 MPI 适配器将 CPU314C－2DP 与编程器（PG）相连，再通电。打开 SIMATIC 管理器，选中站点“SIMATIC 300 站点”，再单击工具栏上的“下载”按钮，如图 3-86 所示，将整个项目（含程序和硬件组态）下载 CPU 中。当合上按钮 SB1 时，灯亮；当合上按钮 SB2 时，灯灭。

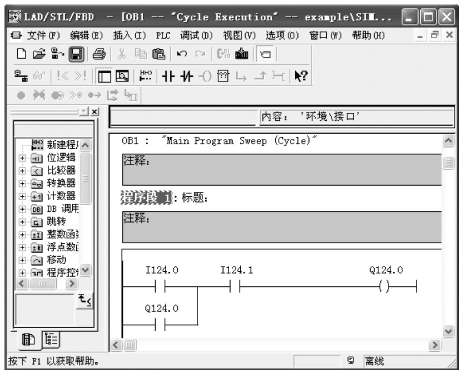



图 3-85 程序

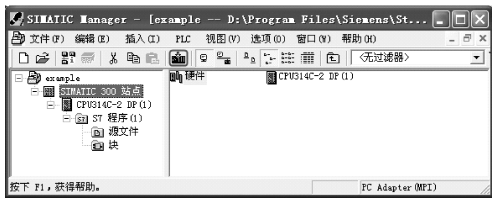



图 3-86 站点下载

【关键点】CPU314C－2DP 在通电状态时，不要插拔 MPI 适配器，这样操作可能会导致 CPU 的 MPI 接口烧毁，应该先断开 CPU 的电源，再插拔 MPI 适配器。

## 3.8 使用帮助

STEP 7 软件有强大的“帮助”功能，早期的版本只有英文版，因此“帮助”全部是英文，而现在，只要安装中文版的 STEP 7 软件，则“帮助”以中文为主，少数模块（例如后安装的 FM355）的“帮助”仍然是英文。学会使用“帮助”是非常重要的，因为 STEP 7 软件的“帮助”信息十分全面，是学习和使用 S7－300/400 PLC 的必备工具。

### 3.8.1 查找关键字或者功能

在工作或者学习时，可以利用“关键字”搜索功能查找帮助信息。以下用一个例子说明查找的方法。

先在 SIMATIC 管理器的单击“帮助”→“目录”，此时弹出帮助信息界面，选中“搜索”选项卡，再在“键入要查找的关键字”输入框中，输入关键字，本例为“OB82”，单击“列出主题”，则有关“OB82”的信息全部显示出来，读者通过阅读这些信息，可了解“OB82”的用法，如图 3-87 所示。

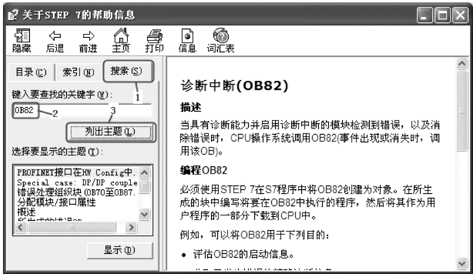



图 3-87 帮助－搜索

### 3.8.2 了解某个逻辑块 FB/SFB/FC/SFC 的功能及引脚的定义

STEP 7 中内置了很多逻辑块（FB/SFB/FC/SFC），在学习和应用时，难免对某个逻辑块有疑问。解决的方法是，在程序编辑器中，找到这个逻辑块，本例为“SFB8”，先选中“SFB8”，如图 3-88 所示，再按键盘上的“F1”，弹出“SFB8”的帮助界面，如图 3-80 所示。也可在程序编辑器中，单击工具栏上的“空逻辑框”按钮，并键入具体逻辑块名（本例为“SFB8”），再按键盘上的“F1”，弹出“SFB8”的帮助界面，如图 3-89 所示。

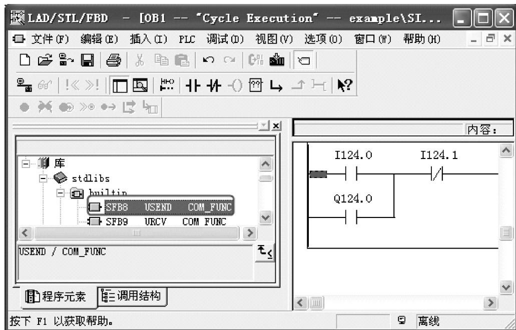



图 3-88 选中逻辑块

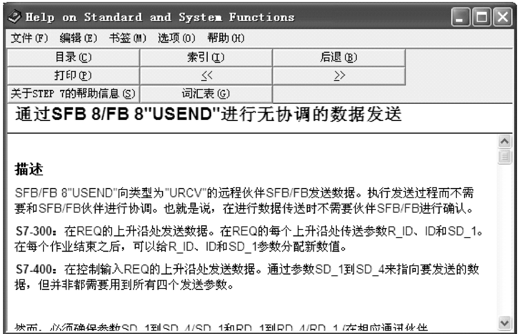



图 3-89 帮助界面

# 第 4 章　S7－300/400 PLC 的编程语言

本章介绍 S7－300/400 PLC 的编程基础知识、指令系统、功能、功能块、系统功能、系统功能块、数据块和组织块。本章内容多，而且难点多，是学习 PLC 入门的关键。

## 4.1　S7－300/400 PLC 的编程基础知识

### 4.1.1　编程元件

S7－300/400 PLC 内部有许多编程元件。编程元件通常指的是硬件，是 PLC 内部具有一定功能的器件总称，这些器件由电子电路、寄存器及存储单元等组成。例如，输入继电器由输入电路和输入映像寄存器构成，输出继电器由输出电路和输出存储映像寄存器构成。定时器和计数器等由特定功能的寄存器构成。为了把这种继电器与传统的电气控制电路中的继电器区分开来，有时也称之为软继电器。这些软继电器的存储区及功能见表 4-1。

表 4-1　存储区及功能

| 地址存储区 | 范　　围 | S7 符号 | 举　　例 | 功 能 描 述 |
|---|---|---|---|---|
| 过程映像输入区 | 输入（位） | I | I0.0 | 扫描周期期间，CPU 从模块读取输入，并记录该区域中的值 |
| | 输入（字节） | IB | IB0 | |
| | 输入（字） | IW | IW0 | |
| | 输入（双字） | ID | ID0 | |
| 过程映像输出区 | 输出（位） | Q | Q 0.0 | 扫描周期期间，程序计算输出值并将它放入此区域，扫描结束时，CPU 发送计算输出值到输出模块 |
| | 输出（字节） | Q B | Q B0 | |
| | 输出（字） | Q W | Q W0 | |
| | 输出（双字） | Q D | Q D0 | |
| 位存储器 | 位存储器（位） | M | M 0.0 | 用于存储程序的中间计算结果 |
| | 位存储器（字节） | M B | M B0 | |
| | 位存储器（字） | M W | M W0 | |
| | 位存储器（双字） | M D | M D0 | |
| 定时器 | 定时器（T） | T | T0 | 为定时器提供存储空间 |
| 计数器 | 计数器（C） | C | C0 | 为计数器提供存储空间 |
| 共享数据块 | 数据（位） | DBX | DBX 0.0 | 数据块用“OPN DB”打开，可以被所有的逻辑块使用 |
| | 数据（字节） | DBB | DBB0 | |
| | 数据（字） | DBW | DBW0 | |
| | 数据（双字） | DBD | DBD0 | |

（续）

| 地址存储区 | 范　围 | S7 符号 | 举　例 | 功能描述 |
|---|---|---|---|---|
| 背景数据块 | 数据（位） | DIX | DIX 0.0 | 数据块用“OPN DI”打开，数据块包含的信息，可以分配给特定的 FB 或者 SFB |
| | 数据（字节） | DIB | DIB0 | |
| | 数据（字） | DIW | DIW0 | |
| | 数据（双字） | DID | DID0 | |
| 本地数据区 | 本地数据（位） | L | L 0.0 | 当块被执行时，此区域包含块的临时数据 |
| | 本地数据（字节） | LB | L B0 | |
| | 本地数据（字） | LW | LW0 | |
| | 本地数据（双字） | L D | LD0 | |
| 外部设备输入区 | 外部设备输入字节 | PIB | PIB0 | 外围设备输入区允许直接访问中央和分布式的输入模块 |
| | 外部设备输入字 | PIW | PIW0 | |
| | 外部设备输入双字 | PID | PID0 | |
| 外部设备输出区 | 外部设备输出字节 | PQ B | PQ B0 | 外围设备输出区允许直接访问中央和分布式的输入模块 |
| | 外部设备输出字 | PQ W | PQ W0 | |
| | 外部设备输出双字 | PQ D | PQ D0 | |
| 程序 | | FC/FB/SFB/SFC | FC1/FB1/SFB1/SFC2 | FC 分用户、标准和系统功能<br>FB 分用户、标准和系统功能块 |

若要存取存储区的某一位，则必须指定地址，包括存储器标识符、字节地址和位号。图 4-1 是一个位寻址的例子。其中，存储器区、字节地址（I 代表输入，2 代表字节 2）和位地址之间用点号（.）隔开。

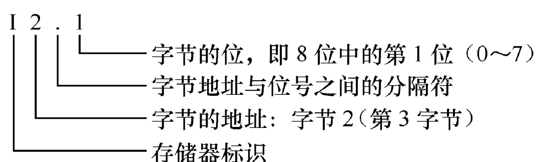



图 4-1　位寻址的例子

## 4.1.2　数制和数据类型

**1. 数制**

PLC 的是一种特殊的计算机，学习计算机必须掌握数制，PLC 更是如此。

（1）二进制

二进制数的 1 位（bit）只能取 0 和 1 两个不同的值，可以用来表示开关量的两种不同的状态，例如触点的断开和接通、线圈的通电和断电以及灯的亮和灭等。在梯形图中，如果该位是 1 可以表示常开触点的闭合和线圈的得电，反之，该位是 0 可以表示常开触点的断开和线圈的断电。二进制用 2#表示，例如 2#1001 1101 1001 1101 就是 16 位二进制常数。十进制的运算规则是逢 10 进 1，二进制的运算规则是逢 2 进 1。

（2）十六进制

十六进制的十六个数字是 0～9 和 A～F（对应于十进制中的 10～15），每个十六进制数字可用 4 位二进制表示，例如 16#A 用二进制表示为 2#1010。B#16#、W#16#和 DW#16#分别表示十六进制的字节、字和双字。十六进制的运算规则是逢 16 进 1。掌握二进制和十六进制之间的转化对于学习西门子 PLC 来说是十分重要的。

（3）BCD 码

BCD 码用 4 位二进制数（或者 1 位十六进制数）表示一位十进制数，例如一位十进制数 9 的 BCD 码是 1001。4 位二进制有 16 种组合，但 BCD 码只用到前十个，而后六个（1010～1111）没有在 BCD 码中使用。十进制的数字转换成 BCD 码是很容易的，例如十进制数 366 转换成十六进制 BCD 码则是 W#16#0366。

**【关键点】** 十进制数 366 转换成十六进制数是 W#16#16E，这是要特别注意的。

BCD 码的最高 4 位二进制数用来表示符号，16 位 BCD 码字的范围是 －999～＋999。32 位 BCD 码双字的范围是 －9999999～＋9999999。不同数制的数的表示方法见表 4-2。

**表 4-2　不同数制的数的表示方法**

| 十进制 | 十六进制 | 二进制 | BCD 码 | 十进制 | 十六进制 | 二进制 | BCD 码 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0000 | 00000000 | 8 | 8 | 1000 | 00001000 |
| 1 | 1 | 0001 | 00000001 | 9 | 9 | 1001 | 00001001 |
| 2 | 2 | 0010 | 00000010 | 10 | A | 1010 | 00010000 |
| 3 | 3 | 0011 | 00000011 | 11 | B | 1011 | 00010001 |
| 4 | 4 | 0100 | 00000100 | 12 | C | 1100 | 00010010 |
| 5 | 5 | 0101 | 00000101 | 13 | D | 1101 | 00010011 |
| 6 | 6 | 0110 | 00000110 | 14 | E | 1110 | 00010100 |
| 7 | 7 | 0111 | 00000111 | 15 | F | 1111 | 00010101 |

**2. 数据类型**

数据是程序处理和控制的对象，在程序运行过程中，数据是通过变量来存储和传递的。变量有两个要素：名称和数据类型。对程序块或者数据块的变量声明时，都要包括这两个要素。

数据的类型决定了数据的属性，例如数据长度和取值范围等。STEP 7 中的数据类型分为 3 大类：基本数据类型、复杂数据类型和参数数据类型。

（1）基本数据类型

基本数据类型是根据 IEC61131－3（国际电工委员会指定的 PLC 编程语言标准）来定义的，每个基本数据类型具有固定的长度且不超过 32 位。

基本数据类型有 12 种，每一种数据类型都具备关键字、数据长度、取值范围和常数表等格式属性。STEP 7 的基本数据类型见表 4-3。

**表 4-3　STEP 7 的基本数据类型**

| 关　键　字 | 长度（位） | 取值范围/格式示例 | 说　　明 |
|---|---|---|---|
| BOOL | 1 | True 或 False(1 或 0) | 布尔变量 |
| BYTE | 8 | B#16#0～B#16#FF(0～255) | 单字节数 |

（续）

| 关 键 字 | 长度(位) | 取值范围/格式示例 | 说 明 |
|---|---|---|---|
| WORD | 16 | 十六进制：W#16#0 ~ W#16#FFFF(0 ~ 65535) | 字(双字节) |
| DWORD | 32 | 十六进制：W#16#0 ~ W#16#FFFF_FFFF(0 ~ 4294967295) | 双字(四字节) |
| CHAR | 8 | “a”、“B”等 | ASCII 字符 |
| INT | 16 | -32768 ~ 32767 | 16 位有符号整数 |
| DINT | 32 | -L#2147483648 ~ L#2147483647 | 32 位有符号整数 |
| REAL | 32 | -3.402823E38 ~ -1.175495E-38<br>+1.175495E-38 ~ +3.402823E38 | IEEE 浮点数 |
| S5TIME | 16 | S5T#0H_0M_0S_0MS ~<br>S5T#2H_46M_30S_0MS | SIMATIC 时间格式 |
| TIME | 32 | -T#24D_20H_31M_23S_648MS ~<br>T#24D_20H_31M_23S_648MS | IEC 时间格式 |
| DATE | 16 | D#1990-1-1 ~ D#2168-12-31 | IEC 日期格式 |
| TIME_OF_DAY | 32 | TOD#0:0:0 ~ TOD#23:59:59.999 | 24 小时时间格式 |

**【关键点】** 有的书上称 WORD（字）为非负整数，而 INT 为整数，即有正负；称 DWORD（双字）为非负双整数，而 DINT 为双整数，即有正负；这非常贴切，也正好解释了字和整数以及双字和双整数的区别。

（2）复杂数据类型

复杂数据类型是一种由其他数据类型组合而成的，或者长度超过 32 位的数据类型，STEP 7 中的复杂数据类型共有 7 类。

1）Date_And_Time（日期时间类型）。其长度为 64 bit（8 B），此数据类型以二进制编码的十进制的格式保存。取值范围是 DT#1990-1-1-0:0:0.0 ~ D#2089-12-31-59:59.999。

2）STRING（字符串）。其长度最多有 254 个字符的组（数据类型 CHAR）。为字符串保留的标准区域是 256 个字节长。这是保存 254 个字符和两个字节的标题所需要的空间。可以通过定义即将存储在字符串中的字符数目来减少字符串所需要的存储空间（例如：string[9]'Siemens'）。

3）ARRAY（数组类型）。定义一个数据类型（基本或复杂）的多维组群。例如：“ARRAY [1..2, 1..3] OF INT”定义 2 × 3 的整数数组。使用下标（“[2,2]”）访问数组中存储的数据。最多可以定义 6 维数组。下标可以是任何整数（-32768 ~ 32767）。

4）STRUCT（结构类型）。该类型是由不同数据类型组成的复合型数据，通常用来定义一组相关数据。例如电动机的一组数据可以按照如下方式定义：

```
Motor: STRUCT
    Speed: INT
    Current: REAL
END_STRUCT
```

5）UDT（用户自定义数据类型）。UDT 是由不同数据类型组成的复合型数据，与 STRUCT 不同的是，UDT 是一个模版，可以用来定义其他的变量。它在 STEP 7 中以块的形式存储，称为 UDT 块。在 S7 的 SIMATIC 管理器中，先选中“块”，再单击菜单栏的“插

入”→“S7 块”→“数据类型”，如图 4-2 所示，弹出数据类型对话框，便可定义新的数据类型。自定义数据类型在后续章节还要介绍。

6）FB 和 SFB（功能块类型）。确定分配的实例数据块的结构，并允许在一个实例 DB 中传送数个 FB 调用的实例数据，在后续章节会重点讲解。

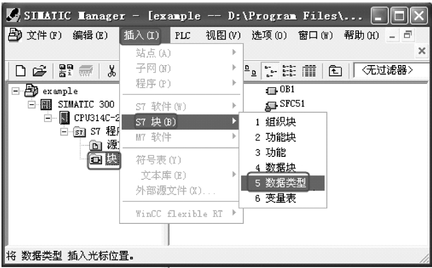


图 4-2 定义 UDT 块的路径

（3）参数数据类型

参数数据类型是一种用于 FC 或者 FB 的参数的数据类型。参数数据类型主要包括以下几种：

- Timer，Counter：定时器和计数器类型。
- BLOCK _ FB，BLOCK _ FC，BLOCK _ DB，BLOCK_SDB：块类型。
- Pointer：6 B 指针类型，传递 DB 块号和数据地址。
- Any：10 B 指针数据类型，传递 DB 块号、数据地址、数据数量以及数据类型。

使用这些参数类型，可以把定时器、计数器、程序块、数据块以及一些不确定类型和长度的数据通过参数传递给 FC 和 FB。参数类型为程序提供了很强的灵活性，可以实现更通用的控制功能。

（4）常数

在 S7－300/400 的许多指令中都用到常数，常数有多种表示方法，如二进制、十进制和十六进制等。在表述二进制和十六进制时，要在数据前分别加“2#”或“16#”格式如下：

二进制常数：2#1100，十六进制常数：16#234B1。其他的数据表述方法举例如下：

ASCII 码：“HELLOW”，实数：－3.1415926，十进制数：234。对于不同品牌的 PLC，常数的表示方法有较大的区别，这点要注意。

几个错误表示方法：八进制的“33”表示成“8#33”，十进制的“33”表示成“10#33”，“2”用二进制表示成“2#2”，这些错误读者要避免。

**【关键点】** 常数可以用二进制、十进制和十六进制表示。为了阅读方便，当用二进制和十六进制表示时，可以在每 4 位之间加下划线，例如 W#16#FFFF_FFFF 和 W#16#FFFFFFFF 实际是相等的。

**【例 4-1】** 请指出以下数据的含义，L#58、S5t#58S、58、C#58、t#58 以及 P#M0.0 BYTE 10。

**解：**

1）L#58：表示双整数 58。

2）S5t#58S：表示 S5 和 S7 定时器中的定时时间 58 s。

3）58：表示整数 58。

4）C#58：表示计数器中的预置值 58。

5）t#58s：表示 IEC 定时器中定时时间 58 s。

6）P#M0.0 BYTE 10：表示从 MB0 开始的 10 个字节。

**【关键点】** 理解【例 4-1】中的数据表示方法至关重要，无论对于编写程序还是阅读程序都是必须要掌握的。

## 4.1.3 寻址方式

**1. 绝对寻址**

要访问一个变量，必须要找到它在存储空间的位置，这个过程就是寻址（Addressing）。在 STEP 7 中，使用地址如 I/O 信号、位内存、计数器、定时器、数据块和功能块都可以通过绝对寻址和符号寻址来访问。

绝对地址是由一个关键字和一个地址数据组成的。STEP 7 中常用的绝对地址关键字见表 4-4。

**表 4-4 绝对地址关键字**

| 关 键 字 | 说 明 | 举 例 |
|---|---|---|
| I/IB/IW/ID | 过程映像区输入信号 | I0.0、IB1、IW2 和 ID2 |
| Q/QB/QW/QD | 过程映像区输出信号 | Q0.0、QB1、QW2 和 QD2 |
| PIB/PIW/PID | 外部设备输入 | PIB1、PIW2 和 PID2 |
| PQB/PQW/PQD | 外部设备输出 | PQB1、PQW2 和 PQD2 |
| M/MB/MW/MD | 位存储区 | M0.0、MB1、MW2 和 MD2 |
| L/LB/LW/LD | 本地数据堆栈区 | L0.0、LB1、LW2 和 LD2 |
| T | 定时器 | T3 |
| C | 计数器 | C5 |
| FC/FB/SFC/SFB | 程序块 | FC1/FB2/SFC3/SFB1 |
| DB | 数据块 | DB1 |

**【关键点】** MD0 由 MB0、MB1、MB2 和 MB3 四个字节组成，MB0 是高字节，而 MB3 是低字节，字节、字和双字的起始地址如图 4-3 所示。

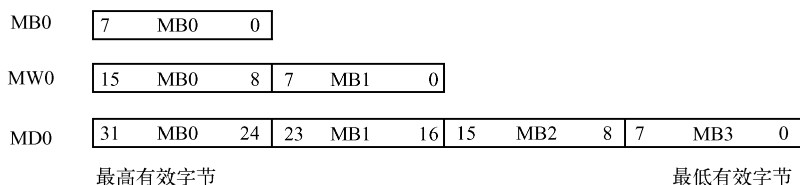



图 4-3 字节、字和双字的起始地址

**【例 4-2】** 如果 MD0 = 16#1F，那么，MB0、MB1、MB2、MB3、M0.0 和 M3.0 的数值是多少？

**解：** 根据图 4-3，MB0 = 0；MB1 = 0；MB2 = 0；MB3 = 16#1F；M0.0 = 0；M3.0 = 1。这点不同于三菱 PLC，读者要注意区分。如不理解此知识点，在编写通信程序时，如 DCS 与 S7 - 300/400 PLC 交换数据时，容易出错。

**【例 4-3】** 如图 4-4 所示的梯形图，是某初学者编写的，请查看有无错误？

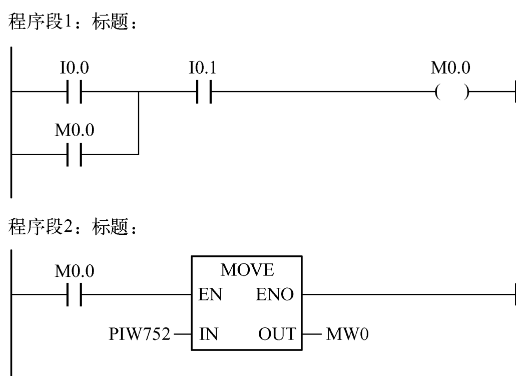



图 4-4　梯形图

**解：**这个程序的逻辑是正确的，但这个程序在实际运行时，并不能采集数据。网路 1 是起停控制，当 M0.0 常开触头闭合后开始采集数据，而且 A－D 转换的结果存放在 MW0 中，MW0 包含两个字节 MB0 和 MB1，而 MB0 包含 8 个位，即 M0.0～M0.7。只要采集的数据经过 A－D 转换，造成 M0.0 位为 0，整个数据采集过程自动停止。初学者很容易犯类似的错误。读者可将 M0.0 改为 M2.0 即可，只要避开 MW0 中包含的 16 个位（M0.0～M0.7 和 M1.0～M1.7）都可行。

**2. 符号寻址**

为变量指定符号名可以简化程序的编写和调试，增加程序的可读性。STEP 7 可以自动将符号地址转化成所需的绝对地址。访问 ARRAY、STRUCT、数据块、本地数据、逻辑块以及用户数据类型（UDT）时，优先选用符号寻址。使用符号寻址前，必须先将符号分配给绝对地址，才能以符号的形式应用它们。

STEP 7 中的符号分为全局符号和局域符号。全局符号是在整个 STEP 7 中可以使用的符号，而局域符号是在某个块中可以使用的符号。全局符号和局域符号的对比见表 4-5。

**表 4-5　全局符号和局域符号的对比**

| | 全局符号 | 局域符号 |
|---|---|---|
| 有效范围 | 在整个程序中有效，可以被所有的块使用，在所有的块中的含义是一样的，整个用户程序中是唯一的 | 只在定义的块中有效，相同的符号在不同的块中，可用于不同的目的 |
| 允许使用的字符 | 字母、数字及特殊字符，除 0x00、0xFF 及引号外的强调号；如用特殊符号，则必须在引号内 | 字母<br>数字<br>下划线（_） |
| 使用独享 | 可以为下列对象定义全局变量：<br>• I/O 信号（如 I、IB、Q、QB、QD 等）<br>• 外部设备 I/O 信号（如 PIB、PQB、PQD 等）<br>• 存储位（如 M、MB、MW、MD 等）<br>• 定时器（T）<br>• 计数器（C）<br>• 程序块（FC/FB/SFC/SFB）<br>• 数据块（DB）<br>• 用户定义数据类型（UDT）<br>• 变量表（VAT） | 可以为下列对象定义局域变量：<br>• 块参数<br>• 块的静态数据<br>• 块的临时数据 |
| 定义位置 | 符号表 | 程序块的变量声明区 |

【例 4-4】将如图 4-5 所示的绝对寻址的起停控制梯形图换成符号寻址梯形图。

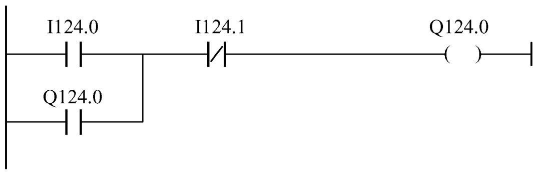



图 4-5　绝对寻址的梯形图

**解**：打开 STEP 7 的 SIMATIC 管理器，先选中“S7 程序（1）”，再双击“符号”，如图 4-6所示，弹出符号编辑器界面，输入如图 4-7 所示的信息，最后单击工具栏的“保存”按钮，将输入的符号分配给相应的地址，例如“起动”分配给地址“I124.0”。再打开程序编辑器，符号寻址的梯形图如图 4-8 所示。

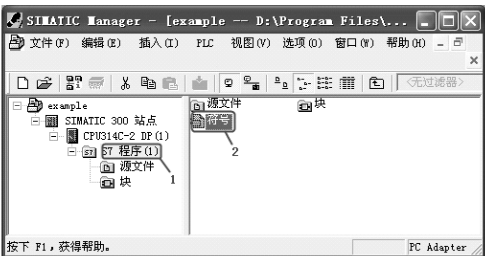



图 4-6　SIMATIC 管理器界面




图 4-7　符号编辑器界面（全局符号）

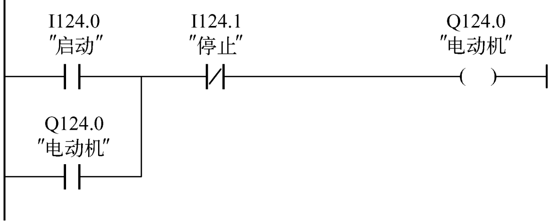



图 4-8　符号寻址的梯形图

如图 4-9 所示，功能块的 IN（输入引脚）上的“Sw_On”和“Sw_Off”数据类型为 BOOL，是局域符号其有效范围仅在 FB1 功能块中。

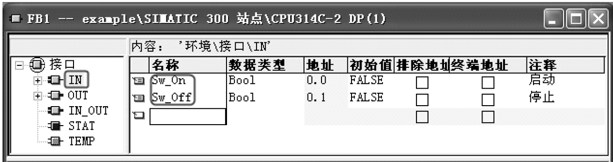



图 4-9　局域符号

**3. 间接寻址**

（1）存储器间接寻址

在存储器间接寻址指令中，给出一个地址指针的存储器，该存储器的内容是操作数所在存储单元的地址。在循环程序中经常用到存储器间接寻址。

地址指针可以是字或双字，定时器（T）、计数器（C）、数据块（DB）、功能块（FB）和功能（FC）的编号范围小于65535，使用字指针就可以。其他地址则要使用双字指针，如果要用双字格式的指针访问一个字、字节或双字存储器，必须保证指针的位编号为0，例如P#Q20.0。

存储器间接寻址的双字指针格式如图4-10所示，其中0～2位为被寻址地址中的位编号，3～18位为寻址字节编号。只有M、L、DB、PI存储区域的双字节才能作地址指针。

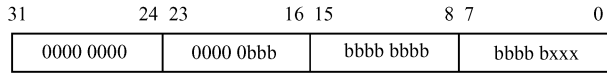



图4-10 存储器间接寻址的双字指针格式

存储器间接寻址应用如下：

L QB[DBD 10]//将输出字节装入累加器1，输出字节的地址指针在数据双字DBD10中，如果DBD10的值为2#0000 0000 0000 0000 0000 0000 0010 0000，装入的是QB4

A M[LD 4]//对存储器位作“与”运算，地址指针在数据双字LD4中，如果LD4的值为2#0000 0000 0000 0000 0000 0000 0010 0011，则是对M4.3进行操作

（2）寄存器间接寻址

地址寄存器AR1和AR2，它们中的内容加上偏移量形成地址指针，指向数值所在的存储单元。寄存器间接寻址中双字指针格式如图4-11所示。

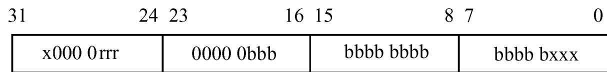



图4-11 寄存器间接寻址的双字指针格式

其中第0～2位（xxx）为被寻址地址中位的编号（0～7），第3～18位为被寻址地址的字节的编号（0～65535），第24～26位（rrr）为被寻址地址的区域标识号，第31位x=0为区域内的间接寻址，第31位x=1为区域间的间接寻址。寄存器间接寻址的区域标识位见表4-6。

**表4-6 寄存器间接寻址的区域标识位**

| 区域标识符 | 存储区 | 位26～24 |
|---|---|---|
| P | 外设输入/输出 | 000 |
| I | 输入过程映像 | 001 |
| Q | 输出过程映像 | 010 |
| M | 位存储区 | 011 |
| DBX | 共享数据块 | 100 |
| DIX | 背景数据块 | 101 |
| L | 块的局域数据 | 111 |

STEP 7 中有两种格式的寄存器间接寻址方式，分别是区域内的间接寻址和区域间的间接寻址。当 31 位为 0 时，为区域内的间接寻址；当 31 位为 1 时，为区域间的间接寻址。

第一种地址指针格式存储区的类型在指令中给出，例如 LDBB[AR1,P#6.0]。在某一存储区内寻址。第 24 ~26 位（rrr）应为 0。

第二种地址指针格式的第 24 ~ 26 位还包含存储区域标识符 rrr，区域间寄存器间接寻址。

如果要用寄存器指针访问一个字节、字或双字，必须保证指针中的位地址编号为 0。

指针常数#P5.0 对应的二进制数为 2#0000 0000 0000 0000 0000 0000 0010 1000。

下面是区内间接寻址的例子：

```
L P#5.0                  //将间接寻址的指针装入累加器 1
LAR1                     //将累加器 1 中的内容送到地址寄存器 1
A M[AR1, P#2.3]          //AR1 中的 P#5.0 加偏移量 P#2.3,实际上是对 M7.3 进行操作
 =Q[AR1, P#0.2]          //逻辑运算的结果送 Q5.2
L DBW[AR1, P#18.0]       //将 DBW23 装入累加器 1
```

下面是区域间间接寻址的例子：

```
L P#M6.0                 //将存储器位 M6.0 的双字指针装入累加器 1
LAR1                     //将累加器 1 中的内容送到地址寄存器 1
T W[AR1, P#50.0]         //将累加器 1 的内容传送到存储器字 MW56
```

P#M6.0 对应的二进制数为 2#1000 0011 0000 0000 0000 0000 0011 0000。因为地址指针 P#M6.0 中已经包含有区域信息，使用间接寻址的指令 T W[AR1,P#50]中没有必要再用地址标识符 M。

### 4.1.4 编程语言

**1. PLC 编程语言的国际标准**

IEC 61131 是 PLC 的国际标准，1992 ~1995 年发布了 IEC 61131 标准中的 1 ~4 部分，我国在 1995 年 11 月发布了 GB/T15969 –1/2/3/4（等同于 IEC 61131 –1/2/3/4）。

IEC 61131 –3 广泛地应用于 PLC、DCS 和工控机、“软件 PLC”、数控系统以及 RTU 等产品。其定义了 5 种编程语言，分别是指令表（Instruction list，IL）、结构文本（Structured text，ST）、梯形图（Ladder diagram，LD）、功能块图（Function block diagram，FBD）和顺序功能图（Sequential function chart，SFC）。

**2. STEP 7 中的编程语言**

STEP 7 中有梯形图、语句表和功能块图 3 种基本编程语言，可以相互转换。通过安装软件包，还有其他的编程语言可供选用，以下简要介绍。

（1）顺序功能图（SFC）

STEP 7 中为 S7 –Graph，它不是 STEP 7 的标准配置，需要安装软件包，S7 –Graph 是针对顺序控制系统进行编程的图形编程语言，特别适合顺序控制程序编写。

（2）梯形图（LAD）

梯形图直观易懂，适合于数字量逻辑控制。梯形图适合于熟悉继电器电路的人员使用。设计复杂的触点电路时最好用梯形图。在小型 PLC 中，其应用最为广泛。

（3）语句表（STL）

语句表的功能比梯形图或功能块图的功能强。语句表可供擅长用汇编语言编程的用户使用。语句表输入快，可以在每条语句后面加上注释。

（4）功能块图（FBD）

“LOGO!”系列微型 PLC 使用功能块图编程。功能块图适合于熟悉数字电路的人员使用。

（5）结构文本（ST）

STEP 7 的 S7－SCL（结构化控制语言）符合 EN61131－3 标准。S7－SCL 适合于复杂的公式计算、复杂的计算任务和最优化算法或管理大量的数据等。S7－SCL 编程语言适合于熟悉高级编程语言（例如 PASCAL 或 C 语言）的人员使用。它不是 STEP 7 的标准配置，需要安装软件包。S7－SCL 应用将越来越广泛。

（6）S7－HiGraph 编程语言

图形编程语言 S7－HiGraph 属于可选软件包，它用状态图（Stategraphs）来描述异步、非顺序过程的编程语言。S7－HiGraph 适合于异步非顺序过程的编程。

（7）S7－CFC 编程语言

可选软件包 S7－CFC（Continuous Function Chart，连续功能图）用图形方式连接程序库中以块的形式提供的各种功能。CFC 适合于连续过程控制的编程。它不是 STEP 7 的标准配置，需要安装软件包。

在 STEP 7 编程软件中，如果程序块没有错误，并且被正确地划分为网络，在梯形图、功能块图和语句表之间可以转换。如果部分网络不能转换，则用语句表表示。

## 4.2　CPU 中的寄存器

**1. 累加器（ACCUx）**

32 位累加器是用于处理字节、字双字的寄存器是执行语句表指令的关键部件。S7－300 有两个累加器（ACCU1 和 ACCU2），S7－400 有 4 个累加器（ACCU1～ACCU4）。几乎所有的语句表的操作都是在累加器中进行的。因此需要用转载指令把操作数送入累加器，在累加器中进行运算和数据处理后，用传送指令把 ACCU1 中的运算结果传送到某个存储单元。

**2. 地址寄存器**

两个 32 位的地址寄存器 AR1 和 AR2 作为指针用于寄存器间接寻址。

**3. 数据块寄存器**

32 位的数据块寄存器 DB 和 DI 的高 16 位分别用来保存打开的共享数据块和背景数据块的编号，低 16 位用于保存打开数据块的字节长度。

**4. 状态字**

状态字是一个 16 位的寄存器，只使用了其中的 9 位，状态字用于存储 CPU 执行后的状态或结果和出错信息。状态字的结构如图 4-12 所示。掌握状态字十分关键，否则是无法阅读语句表程序的，更谈不上编写语句表程序。

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 未用 | BR | CC1 | CC0 | OV | OS | OR | STA | RLO | /FC |

图 4-12　状态字的结构

（1）首次检测位/FC

状态字的第 0 位是首次检测位，该位的状态 0 表示一个梯形逻辑程序的开始，或指令为逻辑串的第一条指令。在逻辑串指令执行过程中该位位 1，输出指令或与 RLO 有关的跳转指令将该位清零，表示一个逻辑串的结束。

（2）逻辑运算结果 RLO

状态字的第 1 位 RLO 是逻辑运算结果。该位用于存储执行位逻辑指令或者比较指令的结果。当 RLO 的状态位 1 时，表示有能流到梯形图的运算点处，为 0 则表示没有能流到该点。如图 4-13 所示的梯形图，可以看到 A 点有能流（实线），而 B 点没有能流（虚线）。

（3）状态位 STA

状态字的第 2 位 STA 是状态位。执行为逻辑指令时，STA 与指令位变量的值一致。可以通过逻辑状态位了解位逻辑指令的状态位。

（4）或位

状态字的第 3 位 OR 是或位。在先逻辑“与”后逻辑“或”的逻辑运算中，OR 位暂存逻辑“与”（串联）的运算结果，以便进行后面的逻辑“或”运算（并联）。输出指令将 OR 位复位，编程时不直接使用 OR 位。

在图 4-13 中梯形图，对应的逻辑代数表达式为：$I0.0 \cdot I0.1 + I0.2 \cdot \overline{I0.3} = Q0.0$，表达式中的“•”表示逻辑“与”，“+”表示逻辑“或”，I0.3 上面的水平线表示“非”运算，等号“=”表示将逻辑运算的结果赋值给 Q0.0。图 4-13 和图 4-14 的运算状态完全相同，只不过图 4-13 是梯形图，而图 4-14 左侧是指令表。指令表中的 A 和 AN 分别表示串联的常开触点和串联的常闭触点，O 表示两个串联电路的并联，等号表示赋值。图 4-14 右边方框中是程序运行时的程序状态结果，其中 STATUS WORD 是状态字。

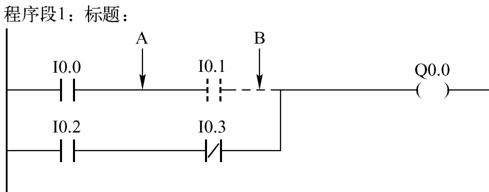



图 4-13　梯形图

**程序段 1：标题：**

```
A    I    0.0
A    I    0.1
O
A    I    0.2
AN   I    0.3
=    Q    0.0
```

| RLO | STA | STANDARD | STATUS WORD |
|---|---|---|---|
| 1 | 1 | 0 | 0_0000_0111 |
| 0 | 0 | 0 | 0_0000_0001 |
| 0 | 1 | 0 | 0_0000_0100 |
| 1 | 1 | 0 | 0_0000_0111 |
| 1 | 0 | 0 | 0_0000_0011 |
| 1 | 1 | 0 | 0_0000_0110 |

图 4-14　语句表程序状态监控

可以看出，当执行最后一条指令后，状态字变成“1_0000_0110”，状态字的第 0 位为“0”，也就是/FC 为 0，当执行其他指令操作时，/FC 为 1,。即在执行完“与”运算和开始下一条“与”运算时，/FC 为 0。

从梯形图，可以看出 STA 表示的是各条指令中布尔变量的值，例如第 5 条指令的 STA 位为 0，表示 I0.3 为 0 状态。

还可以看出 RLO 是逻辑运算的结果，例如执行完第二条指令后，RLO 为 0，原因在于 I0.1 是常开触点，而且没有闭合而造成，如果把 I0.1 闭合则 RLO 随之改变成 1。从梯形图 4-13中还可看出，B 点的运算结果保存在 OR 位中。

（5）溢出位 OV

状态字的第 5 位 OV 是溢出位。如果算数运算或逻辑运算指令执行出错（例如溢出，除数为零等），溢出位被置 1。如果后面影响该位的指令的执行没有出错，该位清零。

（6）溢出状态保持位 OS

状态字的第 4 位 OS 是溢出存储位。OV 位被置 1 时，OS 位也被置 1。OV 位被后面的指令清零后，OS 位仍然保持不变，所以它保存了 OV 位，用于保存前面的指令是否发生过错误。只有 JOS 指令（OS 为 1 时跳转）、块调用指令和块结束指令才能复位 OS 位。

如图 4–15 所示，L 指令将 1100 分别装载到累加器 1 和累加器 2 中，“＊I”是整数乘法指令，运算结果为 1210000 超过了 16 位整数乘法的范围 32767，因此产生溢出，状态字的第 4 和 5 位为 1。在实际工程编程中 OS 位较少应用。

程序段 1：标题：

```
L      1100
L      1100
*I
```

| RLO | STA | STANDARD | STATUS WORD |
|---|---|---|---|
| 0 | 1 | 1100 | 0_0000_0100 |
| 0 | 1 | 1100 | 0_0000_0100 |
| 0 | 1 | 1210000 | 0_1011_0100 |

图 4–15　语句表程序状态监控

（7）条件码 2（CC1）和条件码 1（CC0）

状态字的 7 位和第 1 位称为条件码 1（CC1）和条件码 0（CC0）。这两位综合起来使用，表示累加器 1 中执行的数学运算或字逻辑运算结果与 0 的大小关系，比较指令的执行结果见表 4–7 和表 4–8。移位和循环移位指令移出的位用 CC1 保存。用户程序一般不直接使用条件码。

**表 4–7　算术运算后的 CC1 和 CC0**

| CC1 | CC0 | 算术运算无溢出 | 整数算术运算有溢出 | 浮点数算术运算有溢出 |
|---|---|---|---|---|
| 0 | 0 | = 0 | 整数相加下溢出（负数值过小） | 正数或负数绝对值过小 |
| 0 | 1 | < 0 | 整数相乘下溢出，加减法上溢出（正数过大） | 负数绝对值过大 |
| 1 | 0 | > 0 | 乘法上溢出，加减法下溢出 | 正数上溢出 |
| 1 | 1 | – | 除法或者 MOD 指令的除数为 0 | 非法浮点数 |

**表 4–8　指令执行后的 CC1 和 CC0**

| CC1 | CC0 | 算术运算无溢出 | 移位和循环指令 | 字逻辑指令 |
|---|---|---|---|---|
| 0 | 0 | 累加器 2 = 累加器 1 | 移出位为 0 | 结果为 0 |
| 0 | 1 | 累加器 2 < 累加器 1 | – | – |
| 1 | 0 | 累加器 2 > 累加器 1 | 移出位为 1 | 结果不为 0 |
| 1 | 1 | 非法浮点数 | – | – |

执行完图 4–15 所示的乘法指令“＊I”后，CC1 和 CC0 分别位 1 和 0，表示乘法运算上溢出。

（8）二进制结果位 BR

状态字的 7 位为二进制结果位 BR。在梯形图中，用方框图表示某些指令、功能（FC）和功能块（FB）。如图 4–16 所示，I0. 0 常开触点接通时，能流流到整数除法指令 DIV_I 的数字量输入端 EN，该指令执行，能流用绿色实线表示。

如果指令执行出错（例如除数为 0），能流就在出现错误的除法指令终止，如图 4–17 所示，其 ENO 端没有能流流出。ENO 可以作为下一个方框的 EN 输入，也就是几个方框可以串联，只有前一个方框正确执行，与它连接的后面的程序才能被执行。EN 和 ENO 的操作数均为能流，数据类型为布尔型（BOOL）。

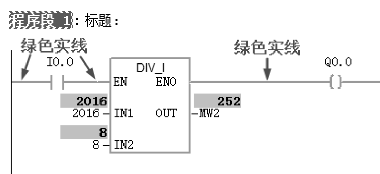



图 4-16　BR 位为 1

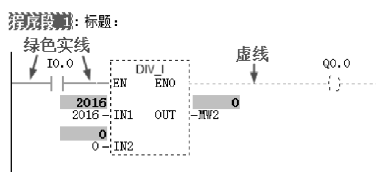



图 4-17　BR 位为 0

用语句表编写的 FB（功能块）和 FC（功能）程序中，必须对 BR 进行管理，当 FB 和 FC 执行无错误时，使 RLO 为 1，并存入 BR，反之将 0 存入 BR 中。可以用 SAVE 指令将 RLO 存入 BR。下面的程序是图 4-16 对应的指令表。

```
       A      I     0.0
       JNB    _001              //如果 I0.0 = 0,则跳转到_001 处
       L      2016              //将 2016 装入累加器 1 的低字节
       L      8                 //将累加器 1 的数据传送到累加器 2,将 8 装入累加器 1
       /I                       //将 2016 除以 8
       T      MW    2           //结果在累加器 1 中,传送到 MW2
       AN     OV                //如果运算没有出错
       SAVE                     //把 RLO 保存到 BR,就是梯形图中 ENO 状态
       CLR                      //清除 RLO
_001:  A      BR
       =      Q     0.0         //用 BR 控制 Q0.0
```

## 4.3　位逻辑指令

位逻辑指令用于二进制数的逻辑运算。位逻辑运算的结果简称为 RLO。

位逻辑指令是最常用的指令之一，主要有与指令、与非指令、或指令、或非指令、置位指令、复位指令和输出指令等。

**1. 触点与线圈**

A（And）：与指令表示串联的常开触点，检测信号 1，与 And 关联。

O（Or）：或指令表示并联的常开触点，检测信号 1，与 Or 关联。

AN（And Not）：与非指令表示串联的常闭触点，检测信号 0，与 And Not 关联。

ON（Or Not）：或非指令表示并联的常闭触点，检测信号 0，与 Or Not 关联。

输出指令“=”将操作结果 RLO 赋值给地址位，与线圈相对应。

与、与非及输出指令示例如图 4-18 所示，图中左侧是梯形图，右侧是与梯形图对应的指令表。当常开触点 I0.0 和常闭触点 I0.2 都接通时，输出线圈 Q0.0 得电（Q0.0 = 1），Q0.0 = 1 实际上就是运算结果 RLO 的数值，I0.0 和 I0.2 是串联关系。

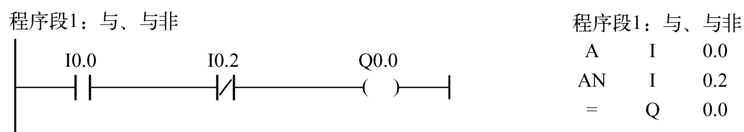



图 4-18　与、与非及输出指令示例

或、或非及输出指令示例如图 4-19 所示，当常开触点 I0.0、常开触点 Q0.0 和常闭触点 M0.0 有一个或多个接通时，输出线圈 Q0.0 得电（Q0.0 = 1），I0.0、Q0.0 和 M0.0 是并

联关系。

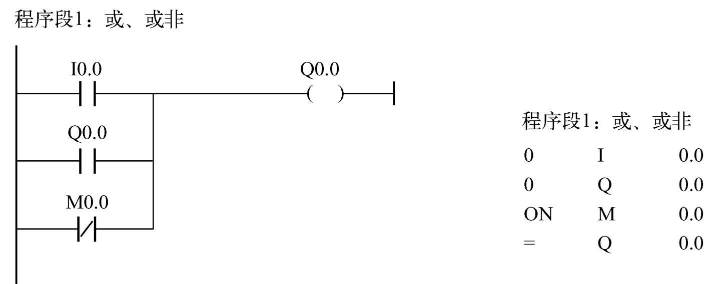



图 4-19　或、或非及输出指令示例

**【例 4-5】** CPU 上电运行后，对 MB0 - MB3 清零复位，请设计梯形图。

**解：** S7 - 300/400 虽然无上电闭合一个扫描周期的特殊寄存器，但有两个方法解决此问题，方法 1 如图 4-20 所示。另一种解法要用到 OB100，将在后续章节讲解。

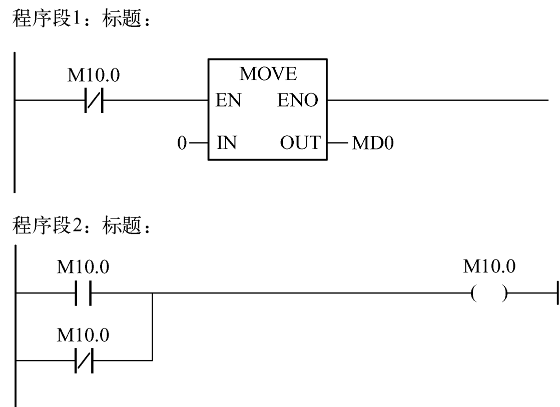



图 4-20　梯形图

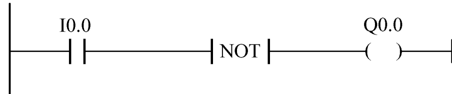
**2. 对 RLO 的直接操作指令**

这类指令可直接对逻辑操作结果 RLO 进行操作，改变状态字中 RLO 的状态。对 RLO 的直接操作指令见表 4-9。

**表 4-9　对 RLO 的直接操作指令**

| 梯形图指令 | STL 指令 | 功 能 说 明 | 说　　明 |
|---|---|---|---|
| --- \| NOT \| --- | NOT | 取反 RLO | 在逻辑串中，对当前 RLO 取反 |
| | SET | 置位 RLO | 将 RLO 置 1 |
| | CLR | 复位 RLO | 将 RLO 清零 |
| -(SAVE) | SAVE | 保存 RLO | 将 RLO 保存到状态字的 BR 位 |

取反触点示例如图 4-21 所示，当 I0.0 为 1 时 Q0.0 为 0，反之当 I0.0 为 0 时 Q0.0 为 1。

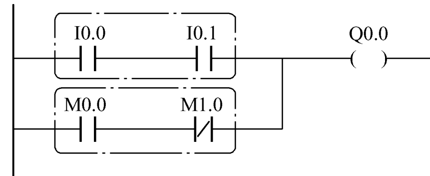



```
程序段1：取反触点
A     I     0.0
NOT
=     Q     0.0
```

图 4-21　取反触点示例

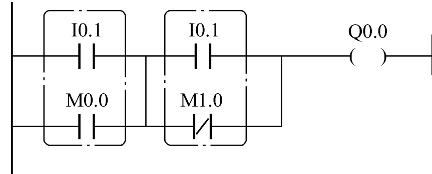
**3. 电路块的串联和并联**

与 S7 -200 PLC 不同，S7 -300/400 PLC 的电路块没有专用的指令。如图 4-22 所示的并联块，实际就是把两个虚线框当作两个块，再将两个块作或运算。如图 4-23 所示的串联块，实际就是把两个虚线框当作两个块，再将两个块作与运算。




```
A    I    0.0
A    I    0.1
O
A    M    0.0
AN   M    1.0
=    Q    0.0
```

图 4-22　并联块示例




```
A(
  O    I    0.0
  O    M    0.0
  )
A(
  O    I    0.1

  ON   M    1.0
  )
  =    Q    0.0
```

图 4-23　串联块示例

**【例 4-6】** 编写程序，实现当压下 SB1 按钮奇数次时，灯亮；当压下 SB1 按钮偶数次时，灯灭。即单键起停控制，请设计梯形图。

**解：** 这个电路是微分电路，但没用到上升沿指令。梯形图如图 4-24 所示。

**4. 复位与置位指令**

S：置位指令将指定的地址位置位（变为 1，并保持）。

R：复位指令将指定的地址位复位（变为 0，并保持）。

如图 4-25 所示为置位/复位指令应用例子，当 I0.0 为 1，Q0.0 为 1，之后，即使 I0.0 为 0，Q0.0 保持为 1，直到 I0.1 为 1 时，Q0.0 变为 0。这两条指令非常有用。

**【关键点】** 置位/复位指令不一定要成对使用。

**【例 4-7】** 用置位/复位指令编写“正转 - 停 - 反转”的梯形图，其中 I0.0 是正转按钮，I0.1 是反转按钮，I0.2 是停止按钮（接常闭触头），Q0.0 是正转输出，Q0.1 是反转输出。

**解：** 梯形图和指令表如图 4-26 所示，可见使用置位/复位指令后，不需要用自锁，程序变得更加简洁。

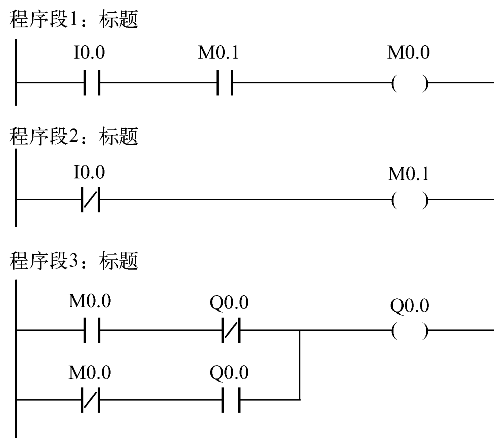



图 4-24　梯形图

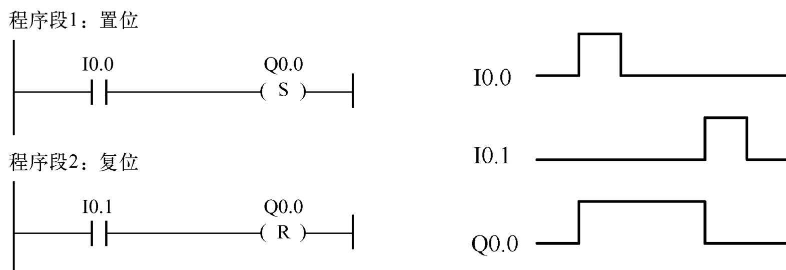



图 4-25　置位/复位指令示例

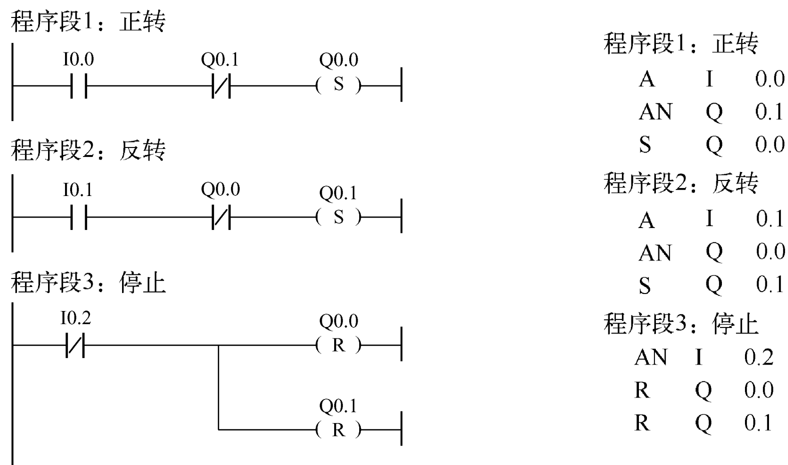



图 4-26　“正转－停－反转”梯形图和指令表

**【例 4-8】** CPU 上电运行后，对 M0.0 置位，并一直保持为 1，请设计梯形图。

**解：** S7－300/400 虽然无上电运行后一直闭合的特殊寄存器，但设计梯形图如图 4-27 所示，可替代此特殊寄存器。

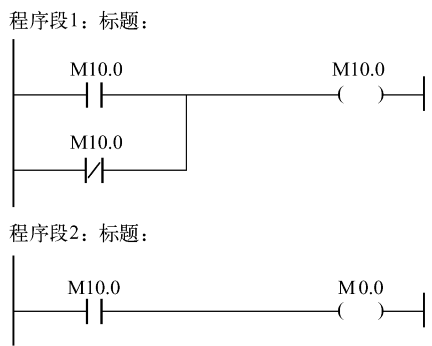



图 4-27　梯形图

**5. RS /SR 双稳态触发器**

RS：置位优先型 RS 双稳态触发器。如果 R 输入端的信号状态为“1”，S 输入端的信号状态为“0”，则复位 RS（置位优先型 RS 双稳态触发器）。否则，如果 R 输入端的信号状态为“0”，S 输入端的信号状态为“1”，则置位触发器。如果两个输入端的 RLO 状态均为“1”，则指令的执行顺序是最重要的。RS 触发器先在指定地址执行复位指令，然后执行置位指令，以使该地址在执行余下的程序扫描过程中保持置位状态。RS /SR 双稳态触发器示例如图 4-28 所示，这个例子的输入与输出的对应关系见表 4-10。

程序段1：置位优先

M0.0　RS　I0.0　R　Q　Q0.0　I0.1　S

程序段2：复位优先

M0.1　SR　I0.2　S　Q　Q0.1　I0.3　R

程序段1：置位优先
```
A   I   0.0
R   M   0.0
A   I   0.1
S   M   0.0
A   M   0.0
=   Q   0.0
```
程序段2：复位优先
```
A   I   0.2
S   M   0.1
A   I   0.3
R   M   0.1
A   M   0.1
=   Q   0.1
```

图 4-28　RS /SR 双稳态触发器示例

**图 4-10　RS /SR 双稳态触发器输入与输出的对应关系**

| 置位优先 RS | | | | 复位优先 SR | | | |
|---|---|---|---|---|---|---|---|
| 输入状态 | | 输出状态 | 说明 | 输入状态 | | 输出状态 | 说明 |
| I0. 0 | I0. 1 | Q0. 0 | 当各个状态断开后，输出状态保持 | I0. 2 | I0. 3 | Q0. 1 | 当各个状态断开后，输出状态保持 |
| 1 | 0 | 0 | | 1 | 0 | 1 | |
| 0 | 1 | 1 | | 0 | 1 | 0 | |
| 1 | 1 | 1 | | 1 | 1 | 0 | |

SR：复位优先型 SR 双稳态触发器。如果 S 输入端的信号状态为“1”，R 输入端的信号状态为“0”，则置位 SR（复位优先型 SR 双稳态触发器）。否则，如果 S 输入端的信号状态

为“0”，R 输入端的信号状态为“1”，则复位触发器。如果两个输入端的 RLO 状态均为“1”，则指令的执行顺序是最重要的。SR 触发器先在指定地址执行置位指令，然后执行复位指令，以使该地址在执行余下的程序扫描过程中保持复位状态。

**【例 4-9】** 设计一个单键启停控制（乒乓控制）的程序，实现用一个单按钮控制一盏灯的亮和灭，即奇数次压下按钮时灯亮，偶数次压下按钮时灯灭。

**解**：先设计其接线图，如图 4-29 所示。

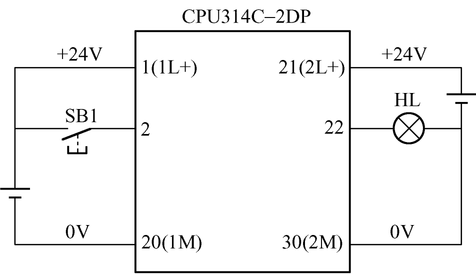



图 4-29　I/O 接线图

梯形图如图 4-30 所示，可见使用 SR 双稳态触发器指令后，不需要用自锁，程序变得更加简洁。当第一次压下按钮时，Q0.0 线圈得电（灯亮），Q0.0 常开触点闭合，当第二次压下按钮时，S 和 R 端子同时高电平，由于复位优先，所以 Q0.0 线圈断电（灯灭）。

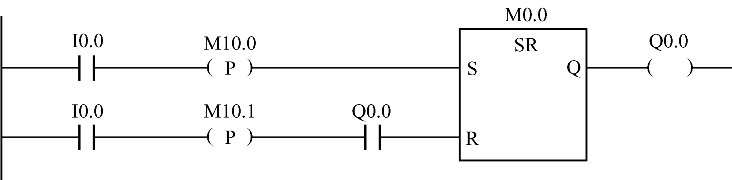



图 4-30　梯形图

这个题目还有另一种解法，就是用 RS 指令，梯形图如图 4-31 所示，当第一次压下按钮时，Q0.0 线圈得电（灯亮），Q0.0 常闭触点断开，当第二次压下按钮时，R 端子高电平，所以 Q0.0 线圈断电（灯灭）。

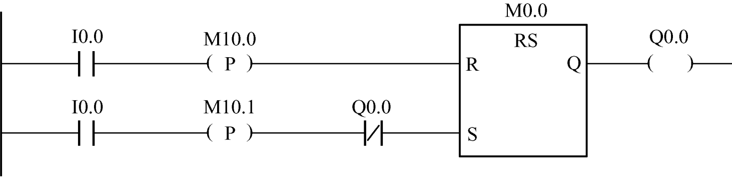



图 4-31　梯形图

**6. 边沿检测指令**

边沿检测指令有负跳沿检测指令（下降沿检测）和正跳沿检测（上升沿检测）指令。

负跳沿检测指令 FN 检测 RLO 从 1 调转到 0 时的下降沿，并保持 RLO＝1 一个扫描周

期。每个扫描周期期间，都会将 RLO 位的信号状态与上一个周期获取的状态比较，以判断是否改变。

下降沿示例的梯形图和指令表如图 4-32 所示，由如图 4-33 所示的时序图可知：当按钮 I0. 0 压下后弹起时，产生一个下降沿，输出 O0. 0 得电一个扫描周期，这个时间是很短的，肉眼是分辨不出来的，因此若 Q0. 0 控制的是一盏灯，肉眼是不能分辨出灯已经亮了一个扫描周期。在后面的章节中多处会用到时序图，请读者务必学会这种表达方式。

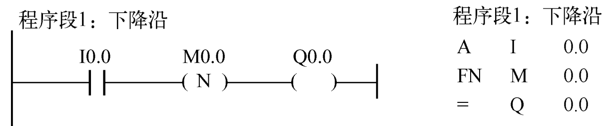



图 4-32　下降沿示例

正跳沿检测指令 FP 检测 RLO 从 0 调转到 1 时的上升沿，并保持 RLO = 1 一个扫描周期。每个扫描周期期间，都会将 RLO 位的信号状态与上一个周期获取的状态比较，以判断是否改变。

上升沿示例的梯形图和指令表如图 4-35 所示，由如图 4-34 所示的时序图可知：当按钮 I0. 0 按下时，产生一个上升沿，输出 O0. 0 得电一个扫描周期，无论按钮闭合多长时间，输出 O0. 0 只得电一个扫描周期。

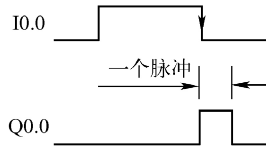



图 4-33　下降沿示例时序图

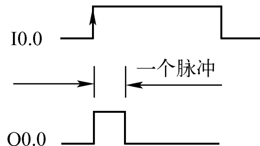



图 4-34　上升沿示例时序图

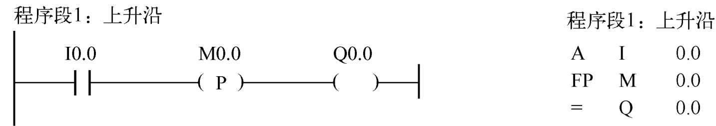



图 4-35　上升沿示例

**【例 4-10】** 边沿检测指令应用梯形图如图 4-36 所示，如果按钮 I0. 0 压下闭合 1 s 后弹起，请分析程序运行结果。

**解：** 时序图如图 4-37 所示，当 I0. 0 压下时，产生上升沿，触点产生一个扫描周期的时钟脉冲，驱动输出线圈 Q0. 1 通电一个扫描周期，Q0. 0 也通电，使输出线圈 Q0. 0 置位，并保持。

当按钮 I0. 0 弹起时，产生下降沿，触点产生一个扫描周期的时钟脉冲，驱动输出线圈 Q0. 2 通电一个扫描周期，使输出线圈 Q0. 0 复位，并保持，Q0. 0 得电共 1 s。

**【例 4-11】** 设计一个程序，实现用一个单按钮控制一盏灯的亮和灭，即按奇数次压下按钮时，灯亮；偶数次压下按钮时，灯灭。

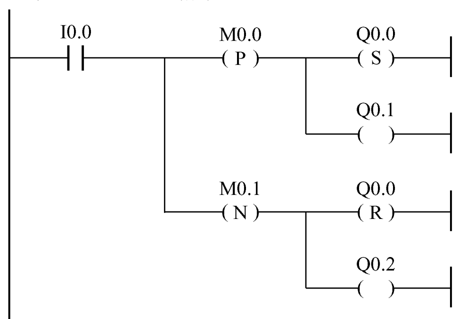



图 4-36　边沿检测指令示例

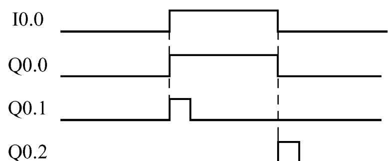
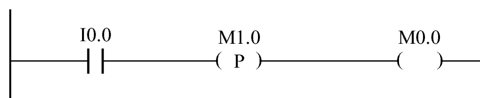



图 4-37　边沿检测指令示例时序图

**解：**当 I0.0 第一次合上时，M0.0 接通一个扫描周期，使得 Q0.0 线圈得电一个扫描周期，当下一次扫描周期到达，Q0.0 常开触点闭合自锁，灯亮。

当 I0.0 第二次合上时，M0.0 线图得电一个扫描周期，使得 M0.0 常闭触点断开，使得灯灭。梯形图如图 4-38 所示。



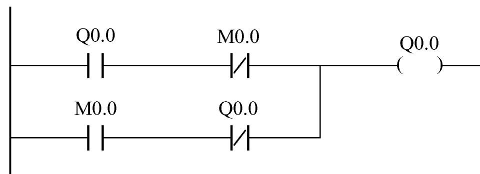




图 4-38　梯形图

**7. 能流取反触点**

能流取反触点的主要作用是把它左边的逻辑运算结果（RLO）取反，如果其左边的运算结果为 1，则变为 0，如果其左边的运算结果为 0，则变为 1。NOT 的应用例子如图 4-39所示，当 I0.0 闭合时，NOT 的左侧运算结果为 1，经过 NOT 触点取反后，RLO 变为 0，因此 Q0.0 输出为 0；反之，如果当 I0.0 断开时，NOT 的左侧运算结果为 0，经过 NOT 触点取反后，RLO 变为 1，因此 Q0.0 输出为 1。

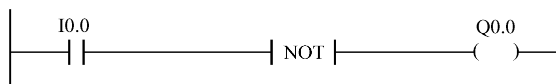



图 4-39　梯形图

**8. 中间输出**

中间输出标有“#”的中间线圈，是一种中间分配单元，用该元件指定的地址来保存它左边电路的逻辑运算结果（RLO），它与其他触点串联，并不影响能流的流向。中间线圈只能放在梯形图的中间，不能放在最左侧，也不能放在最右侧。中间输出的应用如图 4-40 所示，当 I0.0 闭合时，中间线圈 M2.0 左侧的逻辑运算结果 RLO 为 1，因此将运算结果存储在线圈 M2.0 中，从而使得 M2.0 常开触点闭合，因此 Q0.1 线圈也得电。M2.0 线圈并不影响其右侧的逻辑运算，其能流流到线圈 M2.0 右侧和 I0.2 左侧之间，Q0.0 是否得电并不取决于线圈 M2.0。

**9. SAVE 指令**

SAVE 指令就是将 RLO 保存到状态字的 BR 位，在下一段程序中，BR 位的状态将参与“与”逻辑运算。在推出逻辑块之前，通过 SAVE 指令，使 BR 位对应的使能输出 ENO 被设

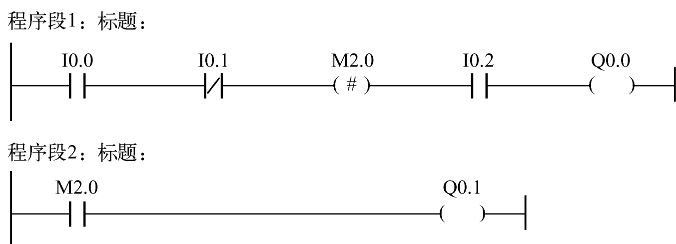



图 4-40　梯形图

置为 RLO 位的值，可以用于块的错误检查。SAVE 指令较为常用，只出现在语句表程序中，但有的梯形图中包含 SAVE 指令，但不显示出来。

**10. SET 和 CLR 指令**

SET 和 CLR 指令无条件将 RLO（逻辑运算结果）置位和复位，随后将 SET 和 CLR 指令后面的赋值指令中的地址变为 1 状态或者 0 状态。SET 和 CLR 指令比较常用，如在初始化组织块 OB100 中，用下列的语句表进行初始化。

```
SET                      //将 RLO 置位
 =     Q      0.0        //将 Q0.0 置位
CLR                      //将 RLO 复位
 =     Q      0.1        //将 Q0.1 复位
```

## 4.4　定时器与计数器指令

### 4.4.1　定时器

STEP 7 的定时器指令相当于继电器接触器控制系统的时间继电器的功能。定时器的数量随 CPU 的类型不同，一般而言可以满足用户使用。

**1. 定时器的种类**

STEP 7 的定时器指令较为丰富，除了常用的接通延时定时器（SD）和断开延时定时器（SF）以外，还有脉冲定时器（SP）、扩展脉冲定时器（SE）和保持型接通延时定时器（SS）共 5 类。

**2. 定时器的使用**

定时器有其存储区域，每个定时器有一个 16 位的字和一个二进制的值。定时器的字存放当前定时值。二进制的值表示定时器的接点状态。

（1）起动和停止定时器

在梯形图中，定时器的 S 端子可以使能定时器，而定时器的 R 端子可以复位定时器。

（2）设定时器的定时时间

STEP 7 中的定时时间由时基和定时值组成，定时时间为时基和定时值的乘积，例如定时值为 1000，时基为 0.01 s，那么定时时间就是 10 s，很多 PLC 的定时都是采用这种方式。定时器开始工作后 ，定时值不断递减，递减至零，表示时间到，定时器会相应动作。

定时器字的格式如图4-41 所示，其中第12 和13 位（即m 和n）是定时器的时基代码，时基代码的含义见表4-11。定时的时间值以3 位 BCD 码格式存放，位于0 ~11（即a ~l），范围为0 ~999。第14 位和15 位不用。

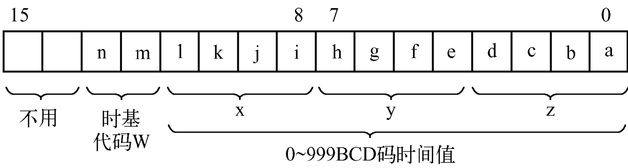



图4-41　定时器字的格式

定时时间有两种表达方式，十六进制数表示和S5 时间格式表示。前者的格式为：W#16#wxyz，其中w 是时间基准代码，xyz 是BCD 码的时间值。例如时间表述为：W#16#1222，则定时时间为222 ×0. 1s =22. 2 s。

**表4-11　时基与定时范围对应表**

| 时基二进制代码 | 时基 | 分辨率（S） | 定 时 范 围 |
|---|---|---|---|
| 00 | 10 ms | 0. 01 | 10 ms ~ 9 s_990 ms |
| 01 | 100 ms | 0. 1 | 100 ms ~ 1 m_39 s_900 ms |
| 10 | 1 s | 1 | 1 s ~ 16 m_39 s |
| 11 | 10 s | 10 | 10 s ~ 2 h_46 m_30 s |

S5 时间格式为：S5T#aH_bM_cS_dMS，其中a 表示小时，b 表示分钟，c 表示秒钟，d 表示毫秒，含义比较明显。例如S5T#1H_2M_3S 表示定时时间为1 小时2 分3 秒。这里的时基是PLC 自动选定的。

**3. 脉冲时间定时器（SP）**

SP：产生指定时间宽度脉冲的定时器。当逻辑位有上升沿时，脉冲定时器指令起动计时，同时节点立即输出高电平“1”，直到定时器时间到，定时器输出为“0”。脉冲时间定时器可以将长信号变成指定宽度的脉冲。如果定时时间未到，而逻辑位的状态变成“0”时，定时器停止计时，输出也变成低电平。脉冲的定时器线圈指令和参数见表4-12。

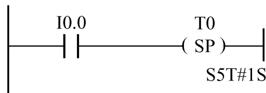
**表4-12　脉冲定时器线圈指令和参数**

| LAD | 参　　数 | 数 据 类 型 | 存　储　区 | 说　　明 |
|---|---|---|---|---|
| T no.<br>-（SP） | T no. | TIMER | T | 表示要起动的定时器号 |
| | 时间值 | S5TIME | I、Q、M、D、L | 定时器时间值 |

用一个例子说明脉冲定时器的使用，梯形图如图4-42 所示，对应的时序图如图4-43 所示，可以看出当I0. 0 接通的时间长于1 s，Q0. 0 输出1 的时间是1 s，而当I0. 0 接通的时间为0. 5 s（小于1 s）时，Q0. 0 输出1 的时间是0. 5 s，无论I0. 0 是否接通，只要I0. 1 接通时，定时器复位，Q0. 0 输出为0。

STEP 7 除了提供脉冲的定时器线圈指令外，还提供更加复杂的方框指令来实现相应的定时功能。脉冲定时器方框指令和参数见表4-13。

程序段1：标题：




程序段2：标题：




程序段3：标题：

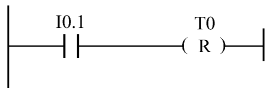



图 4-42　脉冲定时器示例

```
程序段1：标题：
  A     I    0.0
  L     S5 T#1S
  SP    T    0
程序段2：标题：
  A     T    0
  =     Q    0.0
程序段3：标题：
  A     I    0.1
  R     T    0
```

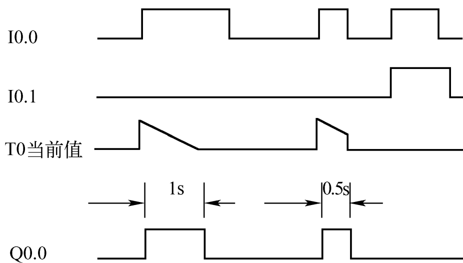
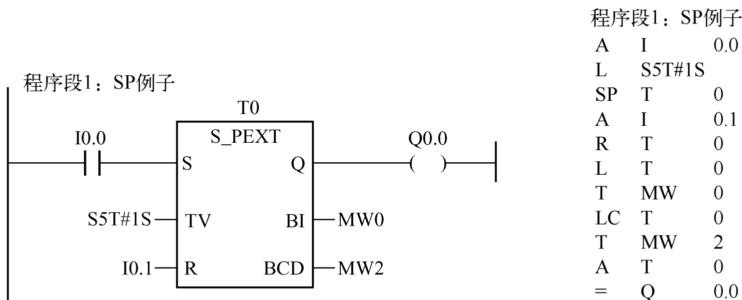



图 4-43　脉冲定时器示例的时序图

**表 4-13　脉冲定时器方框指令和参数**

| LAD | 参　　数 | 数 据 类 型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| T no.<br>S_PULSE<br>S　Q<br>TV　BI<br>R　BCD | T no. | TIMER | 要起动的计时器号，如 T0 | T |
| | S | BOOL | 起动输入端 | I、Q、M、D、L |
| | TV | S5TIME | 定时时间（S5TIME 格式） | |
| | R | BOOL | 复位输入端 | |
| | Q | BOOL | 定时器的状态 | |
| | BI | WORD | 当前时间（整数格式） | |
| | BCD | WORD | 当前时间（BCD 码格式） | |

脉冲定时器方框指令的示例如图 4-44 所示。




```
程序段1：SP例子
A     I     0.0
L     S5T#1S
SP    T     0
A     I     0.1
R     T     0
L     T     0
T     MW    0
LC    T     0
T     MW    2
A     T     0
=     Q     0.0
```

图 4-44　脉冲定时器方框指令示例

### 4. 扩展脉冲时间定时器（SE）

扩展脉冲时间定时器（SE）和脉冲时间定时器（SP）指令相似，但 SE 指令具有保持功能。扩展脉冲时间定时器的线圈指令和参数见表 4-14。

**表 4-14　扩展脉冲定时器线圈指令和参数**

| LAD | 参　　数 | 数 据 类 型 | 存　储　区 | 说　　明 |
|---|---|---|---|---|
| T no.<br>-(SE) | T no. | TIMER | T | 表示要起动的定时器号 |
| | 时间值 | S5TIME | I、Q、M、D、L | 定时器时间值 |

用一个例子来说明 SE 线圈指令的使用，梯形图和指令表如图 4-45 所示，对应的时序图如图 4-46 所示。当 I0.0 有上升沿时，定时器 T0 起动，同时 Q0.0 输出高电平“1”，定时时间到后，输出自动变为“0”（尽管此时 I0.0 仍然闭合），当 I0.0 有上升沿时，且闭合时间没有到定时时间，Q0.0 仍然输出为“1”，直到定时时间到为止。无论什么情况下，只要复位输入端起作用，本例为 I0.1 闭合，则定时器复位，输出为“0”。

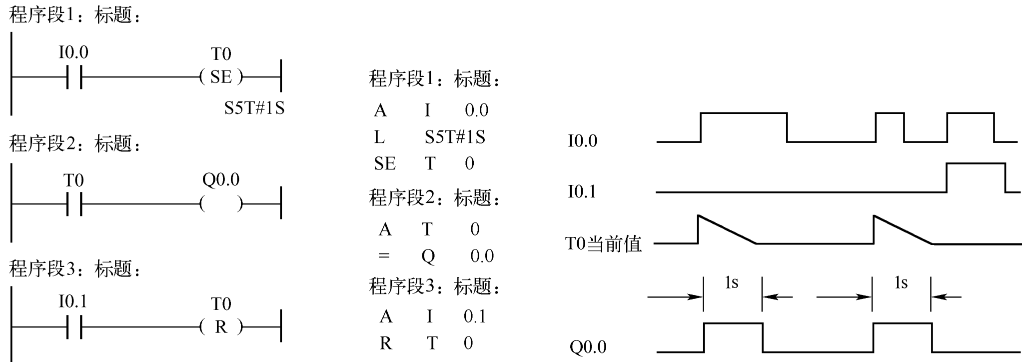



图 4-45　扩展脉冲定时器示例　　　　图 4-46　扩展脉冲定时器示例的时序图

STEP 7 除了提供扩展脉冲的定时器线圈指令外，还提供了更加复杂的方框指令来实现相应的定时功能。扩展脉冲定时器方框指令和参数见表 4-15。

**表 4-15　扩展脉冲定时器方框指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| T no.<br>S_PEXT<br>S　Q<br>TV　BI<br>R　BCD | T no. | TIMER | 要起动的定时器号，如 T0 | T |
| | S | BOOL | 起动输入端 | I、Q、M、D、L |
| | TV | S5TIME | 定时时间（S5TIME 格式） | |
| | R | BOOL | 复位输入端 | |
| | Q | BOOL | 定时器的状态 | |
| | BI | WORD | 当前时间（整数格式） | |
| | BCD | WORD | 当前时间（BCD 码格式） | |

扩展脉冲定时器方框指令的示例如图 4-47 所示。

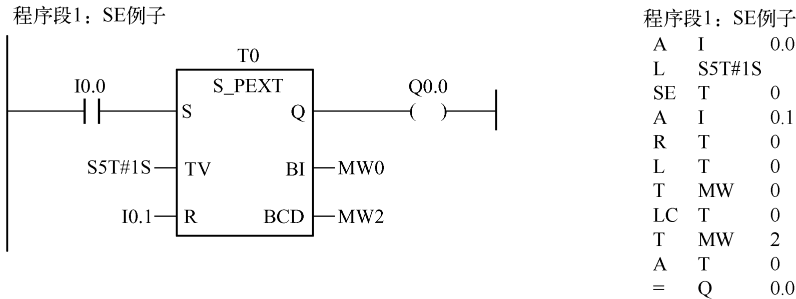



图 4-47　扩展脉冲定时器方框指令示例

#### 5. 接通延时定时器（SD）

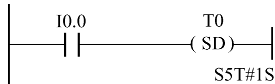
接通延时定时器（SD）相当于继电器接触器控制系统中的通电延时时间继电器。通电延时继电器的工作原理是：线圈通电，触点延时一段时间后动作。SD 指令是当逻辑位接通时，定时器开始定时，计时过程中，定时器的输出为“0”，定时时间到，输出为“1”，整个过程中，逻辑位要接通，只要逻辑位断开，则输出为“0”。接通延时定时器最为常用。接通延时定时器的线圈指令和参数见表 4-16。

**表 4-16　接通延时定时器线圈指令和参数**

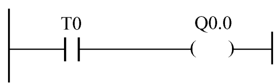
| LAD | 参　数 | 数据类型 | 存　储　区 | 说　明 |
|---|---|---|---|---|
| T no.<br>–（SD） | T no. | TIMER | T | 表示要起动的定时器号 |
| | 时间值 | S5TIME | I、Q、M、D、L | 定时器时间值 |

用一个例子来说明 SD 线圈指令的使用，梯形图和指令表如图 4-48 所示，对应的时序图如图 4-49 所示。当 I0.0 闭合时，定时器 T0 开始定时，定时 1s 后（I0.0 一直闭合），Q0.0 输出高电平“1”，若 I0.0 的闭合时间不足 1s，Q0.0 输出为“0”，若 I0.0 断开，Q0.0 输出为“0”。无论什么情况下，只要复位输入端起作用，本例为 I0.1 闭合，则定时器复位，Q0.0 输出为“0”。

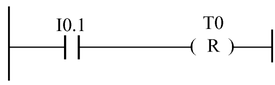
程序段1：标题：




程序段2：标题：




程序段3：标题：




```
程序段1
 A    I     0.0
 L    S5T#1S
 SD   T     0
程序段2
 A    T     0
 =    Q     0.0
程序段3
 I    I     0.1
 T    T     0
```

图 4-48　接通延时定时器示例

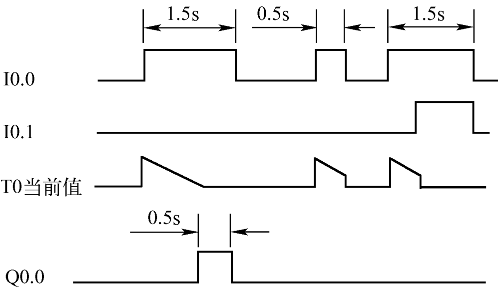



图 4-49　接通延时定时器示例的时序图

STEP 7 除了提供接通延时定时器线圈指令外，还提供更加复杂的方框指令来实现相应的定时功能。接通延时定时器方框指令和参数见表 4-17。

**表 4-17　接通延时定时器方框指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| T no.<br>S_ODT<br>S　Q<br>TV　BI<br>R　BCD | T no. | TIMER | 要起动的定时器号，如 T0 | T |
| | S | BOOL | 起动输入端 | I、Q、M、D、L |
| | TV | S5TIME | 定时时间（S5TIME 格式） | |
| | R | BOOL | 复位输入端 | |
| | Q | BOOL | 定时器的状态 | |
| | BI | WORD | 当前时间（整数格式） | |
| | BCD | WORD | 当前时间（BCD 码格式） | |

接通延时定时器方框指令的示例如图 4-50 所示。

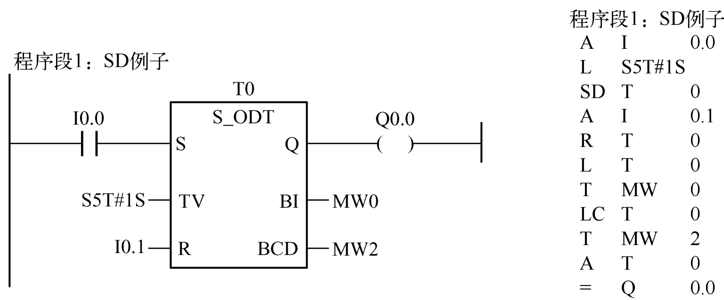



图 4-50　接通延时定时器方框指令示例

【**例 4-12**】设计一段程序，实现一盏灯亮 3 s，灭 3 s，不断循环，且能实现起停控制。

**解**：接线图如图 4-51 所示，梯形图如图 4-52 所示。这个梯形图虽然比较简单，但初学者往往不易看懂。控制过程是：当 SB1 合上，定时器 T0 定时 3 s 后 Q0.0 控制的灯灭，与此同时定时器 T1 起动定时，3 s 后，T1 的常闭触点断开切断 T0，进而 T0 的常开触点切断 T1；此时 T1 的常闭触点闭合 T0 又开始定时，Q0.0 灯亮，如此周而复始，Q0.0 控制灯闪烁。

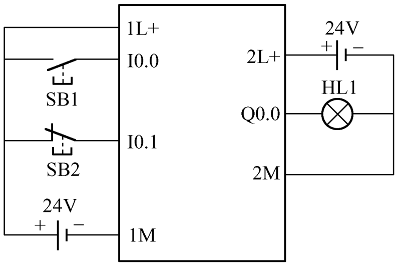



图 4-51　接线图

本例的第二种解法如图 4-53 所示，定时器用方框图表示，这种解法更容易理解。

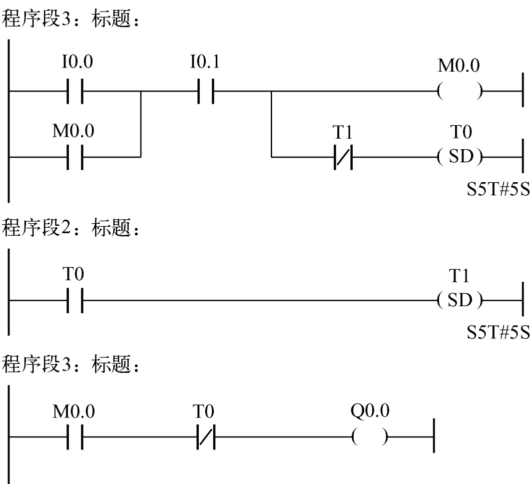
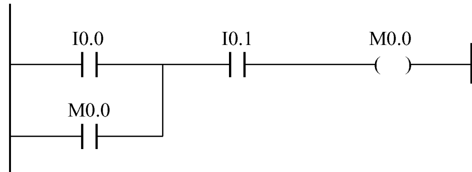



图 4-52　梯形图

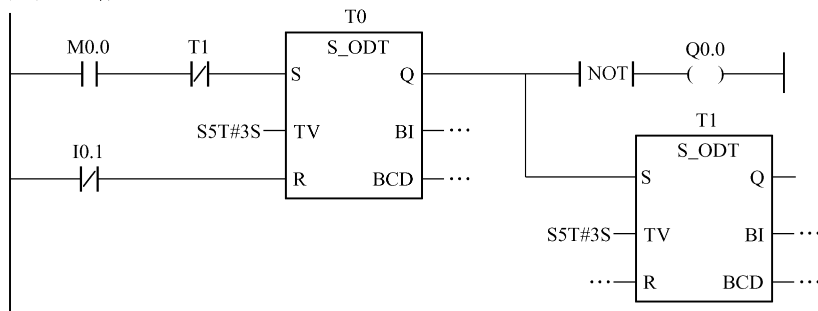
**6. 保持型接通延时定时器（SS）**

保持型接通延时定时器（SS）与接通延时定时器（SD）类似，但 SS 定时器具有保持功能。一旦逻辑位有上升沿发生，定时器起动计时，延时时间到，输出高电平“1”，即使逻辑位为“0”也不影响定时器的工作。必须用复位指令才能使定时器复位。保持型接

日 程序段1：标题：




日 程序段2：标题：




日 程序段3：标题：

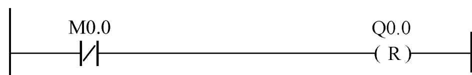



图 4-53　梯形图

通延时定时器的线圈指令和参数见表 4-18。

**表 4-18　保持型接通延时定时器线圈指令和参数**

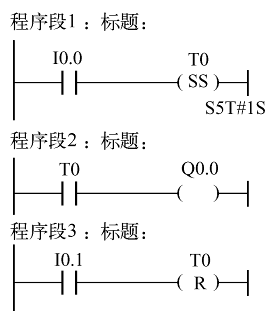
| LAD | 参　　数 | 数 据 类 型 | 存　储　区 | 说　　明 |
|---|---|---|---|---|
| T no.<br>-（SS） | T no. | TIMER | T | 表示要起动的定时器号 |
| | 时间值 | S5TIME | I、Q、M、D、L | 定时器时间值 |

用一个例子来说明 SS 线圈指令的使用，梯形图和指令表如图 4-54 所示，对应的时序图如图 4-55 所示。当 I0. 0 闭合产生一个上升沿时，定时器 T0 开始定时，定时 1 s 后（无论 I0. 0 是否闭合），Q0. 0 输出为高电平“1”，直到复位有效为止，本例为 I0. 1 闭合产生上升沿，定时器复位，Q0. 0 输出为低电平“0”。

STEP 7 除了提供保持型接通延时定时器线圈指令外，还提供更加复杂的方框指令来实现相应的定时功能。保持型接通延时定时器方框指令和参数见表 4-19。

**表 4-19　保持型接通延时定时器方框指令和参数**

| LAD | 参　　数 | 数 据 类 型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| T no.<br>S_ODTS<br>S　Q<br>TV　BI<br>R　BCD | T no. | TIMER | 要起动的定时器号，如 T0 | T |
| | S | BOOL | 起动输入端 | I、Q、M、D、L |
| | TV | S5TIME | 定时时间（S5TIME 格式） | |
| | R | BOOL | 复位输入端 | |
| | Q | BOOL | 定时器的状态 | |
| | BI | WORD | 当前时间（整数格式） | |
| | BCD | WORD | 当前时间（BCD 码格式） | |




```
程序段1：标题：
A  I  0.0
L  S  5T#1S
S  T  0
程序段2：标题：
A  T  0
=  Q  0.0
程序段3：标题：
A  I  0.1
R  T  0
```

图 4-54　保持型接通延时定时器示例

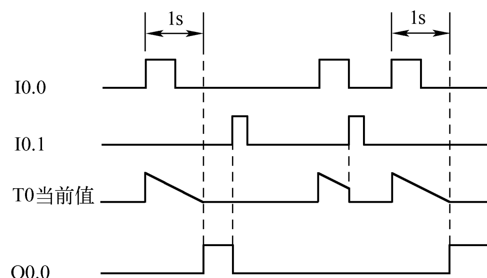



图 4-55　保持型接通延时定时器示例的时序图

保持型接通延时定时器方框指令的示例如图 4-56 所示。

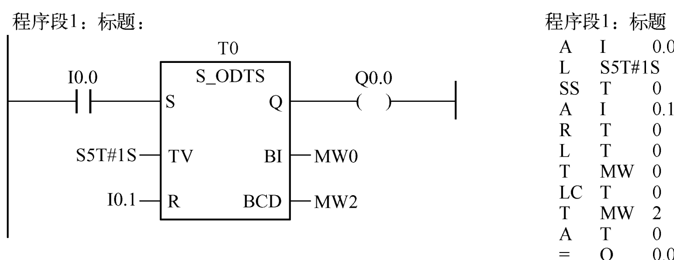



图 4-56　保持型接通延时定时器方框指令的示例

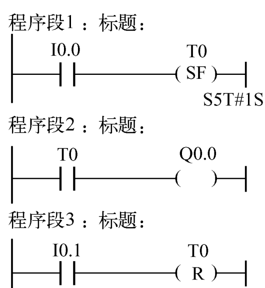
**7. 断开延时定时器（SF）**

断开延时定时器（SF）相当于继电器控制系统的断电延时时间继电器，是定时器指令中唯一一个由下降沿起动的定时器指令。断开延时定时器的线圈指令和参数见表 4-20。

**表 4-20　断开延时定时器线圈指令和参数**

| LAD | 参　　数 | 数 据 类 型 | 存　储　区 | 说　　明 |
|---|---|---|---|---|
| T no.<br>-（SF） | T no. | TIMER | T | 表示要起动的定时器号 |
| | 时间值 | S5TIME | I、Q、M、D、L | 定时器时间值 |

用一个例子来说明 SF 线圈指令的使用，梯形图和指令表如图 4-57 所示，对应的时序图如图 4-58 所示。当 I0.0 闭合时，Q0.0 输出高电平“1”，当 I0.0 断开时产生一个下降沿，定时器 T0 开始定时，定时 1 s 后（无论 I0.0 是否闭合），定时时间到，Q0.0 输出为低电平“0”。任何时候复位有效时，定时器 T0 定时停止，Q0.0 输出为低电平“0”。




```
程序段1：标题：
A   I   0.0
L   S5T#1S
SF  T   0
程序段2：标题：
A   T   0
=   Q   0.0
程序段3：标题：
A   I   0.1
R   T   0
```

图 4-57　断开延时定时器示例

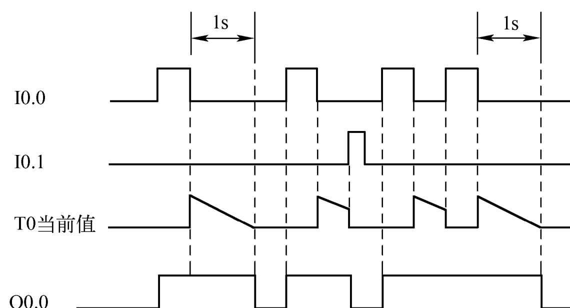



图 4-58　断开延时定时器示例的时序图

STEP 7 除了提供断开延时定时器线圈指令外，还提供更加复杂的方框指令来实现相应的定时功能。断开延时定时器方框指令和参数见表 4-21。

**表 4-21　断开延时定时器方框指令和参数**

| LAD | 参　　数 | 数据类型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| T no.<br>S_OFFDT<br>S　Q<br>TV　BI<br>R　BCD | T no. | TIMER | 要起动的定时器号，如 T0 | T |
| | S | BOOL | 起动输入端 | I、Q、M、D、L |
| | TV | S5TIME | 定时时间（S5TIME 格式） | |
| | R | BOOL | 复位输入端 | |
| | Q | BOOL | 定时器的状态 | |
| | BI | WORD | 当前时间（整数格式） | |
| | BCD | WORD | 当前时间（BCD 码格式） | |

断开延时定时器方框指令的示例如图 4-59 所示。

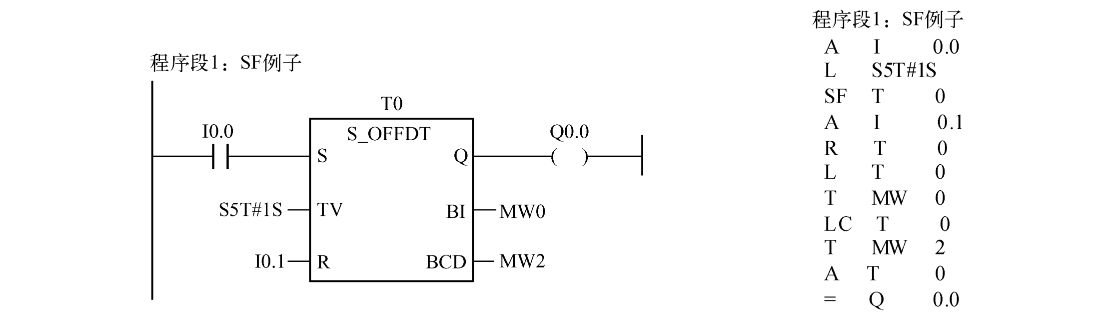



图 4-59　断开延时定时器方框指令的示例

**【例 4-13】** 某车库中有一盏灯，当人离开车库后，压下停止按钮，5 s 后灯熄灭，请编写程序。

**解：** 当接通 SB1 按钮，灯 HL1 亮；按下 SB2 按钮 5 s 后，灯 HL1 灭。接线图如图 4-60 所示，梯形图如图 4-61 所示。

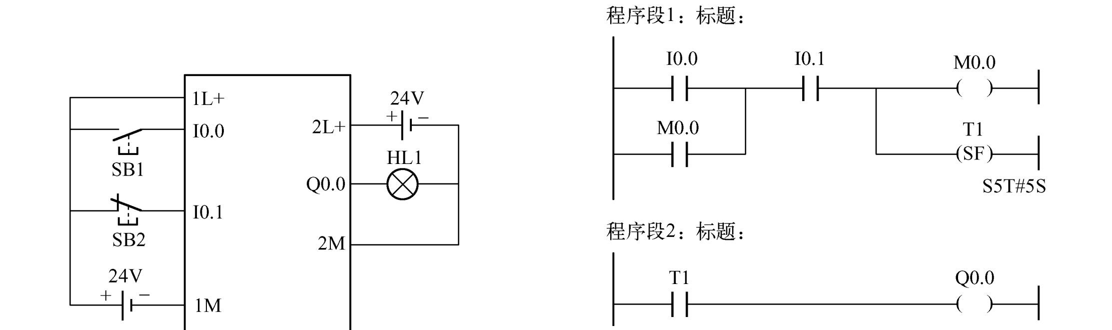



图 4-60　接线图　　　　图 4-61　梯形图

**【例 4-14】** 鼓风机系统一般有引风机和鼓风机两级构成。当压下起动按钮之后，引风机先工作，工作 5 s 后，鼓风机工作。压下停止按钮之后，鼓风机先停止工作，5 s 之后，引风

机才停止工作，请编写程序。

**解：**

（1）PLC 的 I/O 分配

见表 4-22

**表 4-22　PLC 的 I/O 分配表**

| 输　入 | | | 输　出 | | |
|---|---|---|---|---|---|
| 名　称 | 符　号 | 输 入 点 | 名　称 | 符　号 | 输 出 点 |
| 起动按钮 | SB1 | I0. 0 | 鼓风机 | KA1 | Q0. 0 |
| 停止按钮 | SB2 | I0. 1 | 引风机 | KA2 | Q0. 1 |

（2）控制系统的接线

鼓风机控制系统的接线比较简单，如图 4-62 所示。

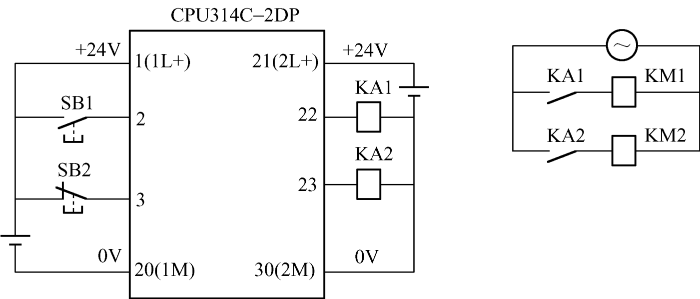



图 4-62　PLC 接线图

（3）编写程序

引风机在按下停止按钮后还要运行 5 s，容易想到要使用 SF 定时器；鼓风机在引风机工作 5 s 后才开始工作，因而容易想到用 SD 定时器，不难设计梯形图，如图 4-63 所示。

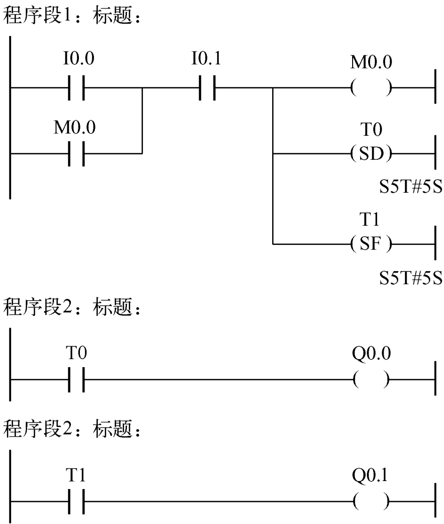



图 4-63　鼓风机控制梯形图

### 4.4.2 IEC 定时器

西门子 PLC 的定时器的数量有限，如果项目较大，定时器不够用时，可以使用 IEC 定时器。IEC 定时器作为系统功能块 SFB 集成在 CPU 的操作系统中。在相应的 CPU 中有以下定时器：脉冲定时器 SFB3（TP）、通电延时定时器 SFB4（TON）和断电定时器 SFB5（TOF）。

**1. 通电延时定时器 SFB4（TON）简介**

通电延时定时器 SFB4（TON）的参数见表 4-23。

**表 4-23　通电延时定时器方框指令和参数**

| LAD | 参　　数 | 数据类型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| C no.<br>S_CD<br>CD　Q<br>S<br>PV　CV<br>CV_BCD<br>R | EN | BOOL | 使能输入端 | |
| | IN | BOOL | 起动定时器 | I、Q、M、D、L、常数 |
| | Q | BOOL | 定时器的状态 | I、Q、M、D、L |
| | PT | TIME | 定时时间 | I、Q、M、D、L、常数 |
| | ET | TIME | 用完的时间 | I、Q、M、D、L |

**2. 应用实例**

下面仅以通电延时定时器 SFB4（TON）为例介绍 IEC 定时器的应用。

**【例 4-15】** 压下按钮 I0.0，3 s 后电动机起动，请设计梯形图。

**解：** 使用 S7 定时器 TON 也可行，以下用 IEC 定时器，梯形图如图 4-64 所示，合并起动定时器，T#3 s 是定时时间，3 s 后 Q0.0 为 1，MD10 中是定时器完成定时时间，本例的时间单位是 0.1 s。

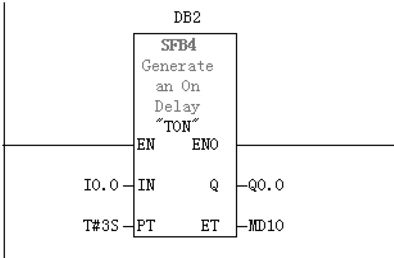


图 4-64　梯形图

### 4.4.3 计数器

计数器的功能是完成计数功能，可以实现加法计数和减法计数，计数范围是 0～999，计数器有 3 种类型：加计数器（S_CU）、减计数器（S_CD）和加减计数器（S_CUD）。

**1. 计数器的存储区**

在 CPU 的存储区中，为计数器保留有存储区。该存储区为每个计数器地址保留一个 16 位的字。计数器的存储格式如图 4-65 所示，其中 BCD 码格式的计数值占用字的 0～11 位，共 12 位，而 12～15 位不使用；二进制格式的计数值占用字的 0～9 位，共 10 位，而 10～15 位不使用。

**2. 加计数器（S_CU）**

加计数器（S_CU）在计数初始值预置输入端 S 上有上升沿时，PV 装入预置值，输入端 CU 每检测到一次上升沿，当前计数值 CV 加 1（前提是 CV 小于 999）；当前计数值大于 0 时，Q 输出为高电平“1”；当 R 端子的状态为“1”时，计数器复位，当前计数值 CV 为

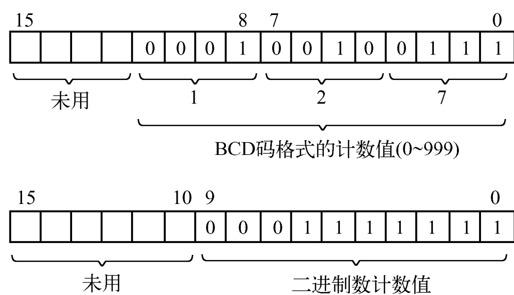



图 4-65　计数器字的格式

“0”，输出也为“0”。加计数器指令和参数见表 4-24。

**表 4-24　加计数器指令和参数**

| LAD | 参　　数 | 数 据 类 型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| C no.<br>S_CD<br>CU　Q<br>S<br>PV　CV<br>CV_BCD<br>R | C no. | COUNTER | 要起动的计数器号，如 C0 | C |
| | CU | BOOL | 加计数输入 | I、Q、M、D、L |
| | S | BOOL | 计数初始值预置输入端 | |
| | PV | WORD | 初始值的 BCD 码 | |
| | R | BOOL | 复位输入端 | |
| | Q | BOOL | 计数器的状态输出 | |
| | CV | WORD | 当前计数值（整数格式） | |
| | CV_BCD | WORD | 当前计数值（BCD 码格式） | |

用一个例子来说明加计数器指令的使用，梯形图和指令表如图 4-66 所示，与之对应的时序图如图 4-67 所示。当 I0.1 闭合时，MW20 将值赋给 PV（假设为 4）；当 I0.0 每产生一个上升沿，计数器 C0 计数 1 次，CV 加 1；只要计数值大于 0，Q0.0 输出高电平“1”。任何时候复位有效时，计数器 C0 复位，CV 清零，Q0.0 输出为低电平“0”。

程序段1：CU例子

C0　S_CU　I0.0 → CU　Q → Q0.0 ( )　I0.1 → S　CV → MW4　MW20 → PV　CV_BCD → MW6　I0.2 → R

程序段1：CU例子

```
A    I    0.0
CU   C    0
BLD  101
A    I    0.1
L    MW   20
S    C    0
A    I    0.2
R    C    0
L    C    0
T    MW   4
LC   C    0
T    MW   6
A    C    0
=    Q    0.0
```

图 4-66　加计数器指令示例

【**关键点**】S7－200 PLC 的增计数器（如 C0），当计数值到预置值时，C0 的常开触点闭合，常闭触点断开，S7－300 PLC 的 S7 计数器无此功能。

### 3. 减计数器（S_CD）

减计数器（S_CD）在计数初始值预置输入端S上有上升沿时，PV装入预置值，输入端CD每检测到一次上升沿，当前计数值CV减1（前提是CV值大于0），当CV等于0时，计数器的输出Q从状态“1”变成状态“0”；当R端子的状态为“1”时，计数器复位，当前计数值为“PV”，输出也为“0”。减计数器指令和参数见表4-25。

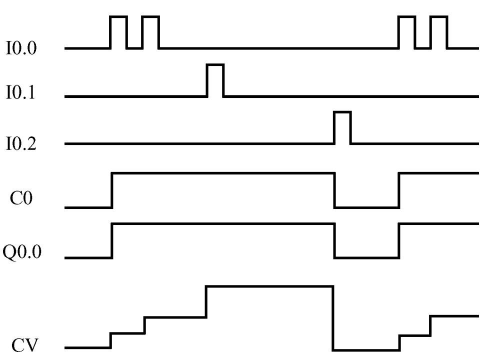
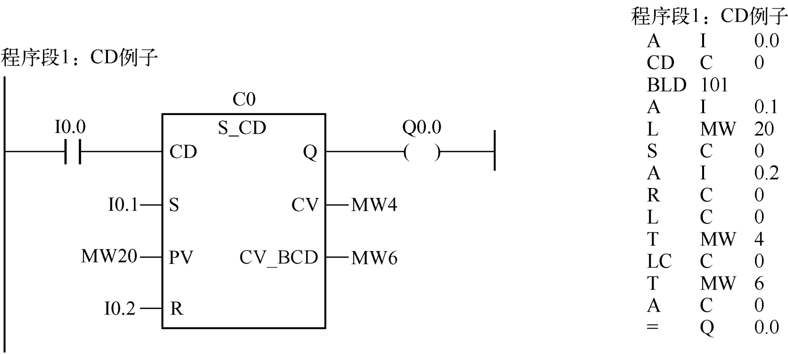



图4-67　加计数器指令示例时序图

表4-25　减计数器指令和参数

| LAD | 参　　数 | 数据类型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| C no.<br>S_CD<br>CD　Q<br>S<br>PV　CV<br>CV_BCD<br>R | C no. | COUNTER | 要起动的计数器号，如C0 | C |
| | CD | BOOL | 减计数输入 | I、Q、M、D、L |
| | S | BOOL | 计数初始值预置输入端 | |
| | PV | WORD | 初始值的BCD码 | |
| | R | BOOL | 复位输入端 | |
| | Q | BOOL | 计数器器的状态输出 | |
| | CV | WORD | 当前计数值（整数格式） | |
| | CV_BCD | WORD | 当前计数值（BCD码格式） | |

用一个例子来说明减计数器指令的使用，梯形图和指令表如图4-68所示，与之对应的时序图如图4-69所示。当I0.1闭合时，MW20将值赋给PV（假设为4），当I0.0每产生一个上升沿，计数器C0计数1次，CV减1，当CV值为0时，Q0.0输出从“1”变成“0”。任何时候复位有效时，定时器C0复位，CV值为0，Q0.0输出为低电平“0”。




```
程序段1：CD例子
A   I   0.0
CD  C   0
BLD 101
A   I   0.1
L   MW  20
S   C   0
A   I   0.2
R   C   0
L   C   0
T   MW  4
LC  C   0
T   MW  6
A   C   0
=   Q   0.0
```

图4-68　加计数器指令示例

【例4-16】设计一个程序，实现用一个单按钮控制一盏灯的亮和灭，即奇数次压下按钮时，灯亮；偶数次压下按钮时，灯灭。

**解：**

当I0.0第一次合上时，M0.0接通一个扫描周期，使得Q0.0线圈得电一个扫描周期，

当下一次扫描周期到达，Q0.0 常开触点闭合自锁，灯亮。

当 I0.0 第二次合上时，M0.0 接通一个扫描周期，C0 计数为 2，Q0.0 线圈断电，使得灯灭，同时计数器复位。梯形图如图 4–70 所示。

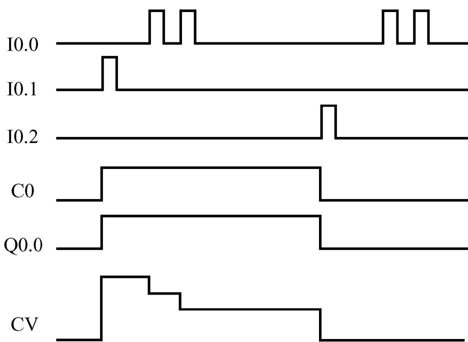



图 4–69　减计数器指令示例时序图

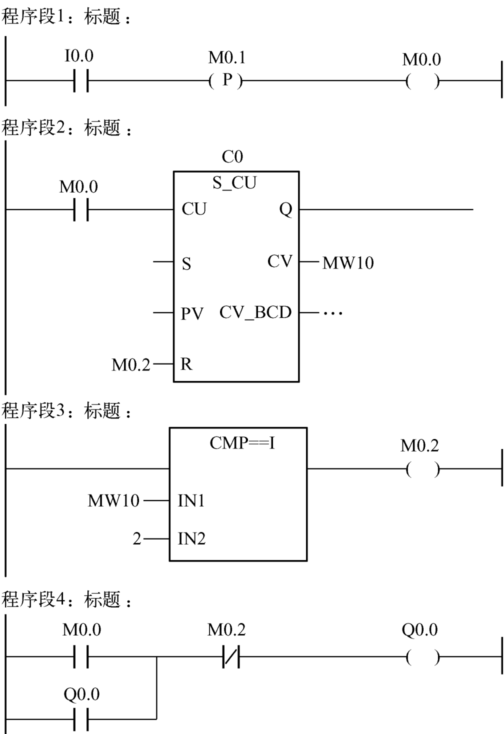



图 4–70　梯形图

**4. 加 – 减计数器（S_CUD）**

加 – 减计数器（S_CUD）在计数初始值预置输入端 S 上有上升沿时，PV 装入预置值，输入端 CD 每检测到一次上升沿，当前计数值 CV 减 1（前提是 CV 值大于 0）；输入端 CU 每检测到一次上升沿，当前计数值 CV 加 1（前提是 CV 值小于 999）；当 CD 和 CU 同时有上升沿时，CV 不变；计数值大于 0 时，计数器的输出 Q 从状态为“1”；计数值等于 0 时，计数器的输出 Q 从状态为“0”；当 R 端子的状态为“1”时，计数器复位，当前计数值为“0”，输出也为“0”。加 – 减计数器指令和参数见表 4–26。

**表 4–26　加 – 减计数器指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| C no.<br>S_CUD<br>CU　Q<br>CD<br>S　CV<br>PV　CV_BCD<br>R | C no. | COUNTER | 要起动的计数器号，如 C0 | C |
| | CD | BOOL | 减计数输入 | I、Q、M、D、L |
| | CU | BOOL | 加计数输入 | |
| | S | BOOL | 计数初始值预置输入端 | |
| | PV | WORD | 初始值的 BCD 码 | |
| | R | BOOL | 复位输入端 | |
| | Q | BOOL | 计数器器的状态输出 | |
| | CV | WORD | 当前计数值（整数格式） | |
| | CV_BCD | WORD | 当前计数值（BCD 码格式） | |

用一个例子来说明加－减计数器指令的使用，梯形图和指令表如图 4-71 所示。当 I0.2 闭合时，MW20 将值赋给 PV（假设为3），当 I0.1 每产生一个上升沿，计数器 C0 计数 1 次，CV 减 1，当 CV 值为 0 时，Q0.0 输出从“1”变成“0”；I0.0 是增计数端。任何时候复位有效时，定时器 C0 复位，CV 值为 0，Q0.0 输出为低电平“0”。

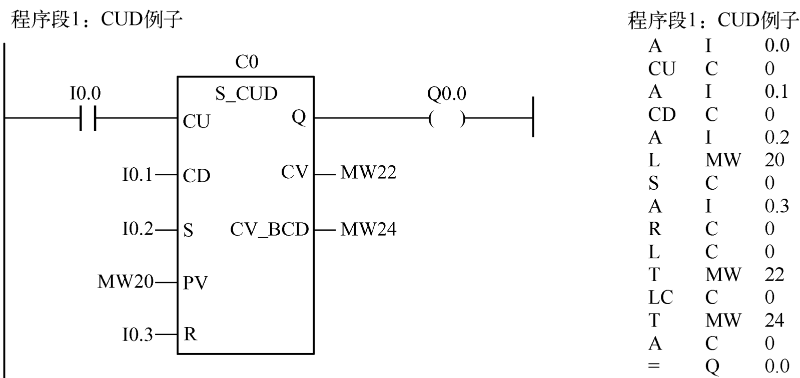



图 4-71　加－减计数器指令示例

**【例 4-17】** 在实际工程应用中，常常在监控面板上使用拨码开关给 PLC 设定数据。I0.0、I0.1、I0.2 对应 SB1、SB2 和 SB3 按钮。当 I0.0 接通 C0 加 1，当 I0.1 接通 C0 减 1，当 I0.2 接通 C0 复位。通过 SB1、SB2 设定 0～9 共 10 个数字。

**解：** 梯形图如图 4-72 所示。

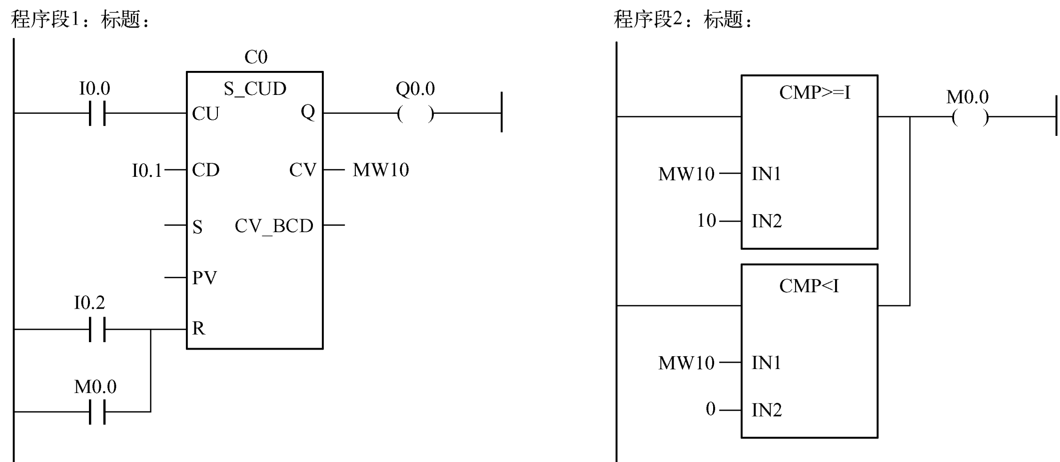



图 4-72　梯形图

## 4.4.4　IEC 计数器

西门子 PLC 的计数器的数量有限，如果大型项目，计数器不够用时，可以使用 IEC 计数器。IEC 计数器作为系统功能块 SFB 集成在 CPU 的操作系统中。在 CPU 中有以下计数器：加计数器 SFB0（CTU）、减计数器 SFB1（CTD）和加减计数器 SFB2（CTUD）。

**1. 加计数器 SFBO（CTU）简介**

加计数器 SFB0（CTU）的参数见表 4-27。

**表 4-27　加计数器 SFB0（CTU）指令和参数**

| LAD | 参数 | 数据类型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| "CTU"<br>EN　ENO<br>CU　Q<br>R　CV<br>PV | EN | BOOL | 使能输入端 | |
| | CU | BOOL | 计数器输入 | I、Q、M、D、L、常数 |
| | R | BOOL | 复位，优先于 CU 端 | I、Q、M、D、L 、常数 |
| | PV | INT | 预设值 | I、Q、M、D、L、常数 |
| | Q | BOOL | 计数器的状态，CV >= PV，Q 输出 1，CV < PV，Q 输出 0 | I、Q、M、D、L |
| | CV | INT | 当前计数值（允许值：0 至 32767） | I、Q、M、D、L |

**2. 应用实例**

以下仅以加计数器 SFB0（CTU）为例介绍 IEC 计数器的应用。

**【例 4-18】** 按下按钮 I0.0，3 次后电动机起动，按下按钮 I0.1 电动机，电动机停止，请设计梯形图。

**解：** 以下用 IEC 计数器，梯形图如图 4-73 所示，I0.0 按下 3 次，MW0 中存储的是当前计数值（CV）为 3，等于预设值（PV），所以 Q0.0 状态变为 1，电动机起动；当按下 I0.1 复位按钮，MW0 中存储的是当前计数值变为 0，小于预设值（PV），所以 Q0.0 状态变为 0，电动机停止。顺便说明此题使用 S7 计数器 CTU 也可行，但程序要复杂一些。

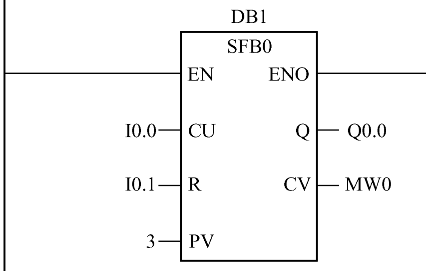



图 4-73　梯形图

# 4.5　数据处理与运算指令

## 4.5.1　装载与传送指令

装载 L 和传送指令 T 用于存储器之间或者存储区和过程输入、输出之间交换数据。装载和传送指令需要累加器的参与。

装载（Load，L）指令将源操作数装入累加器 1，而累加器 1 原有的数据移入累加器 2。装入指令可以对字节（8 位）、字（16 位）、双字（32 位）数据并行操作。

传送（Transfer，T）指令将累加器 1 中的内容写入目的存储区中，累加器 1 的内容不变。

**1. 立即寻址的装载与传送指令**

立即寻址的操作数直接在指令中，下面是使用立即寻址的例子。

```
L    -38          //将 16 位十进制常数 -38 装入累加器 1 的低字 ACCU1 - L
L    L#5          //将 32 位常数 5 装入累加器 1
```

```
L   2#0001_1001_1110_0010     //将 16 位二进制常数装入累加器 1 的低字 ACCU1 - L
L   25.38                     //将 32 位浮点数常数(25.38)装入累加器 1
L   'ABCD'                    //将 4 个字符装入累加器 1
L   TOD#12:30:3.0             //将 32 位实时时间常数装入累加器 1
L   D#2004 - 2 - 3            //将 16 位日期常数装入累加器 1 的低字 ACCU1 - L
L   C#50                      //将 16 位计数器常数装入累加器 1 的低字 ACCU1 - L
L   T#1M20S                   //将 16 位定时器常数装入累加器 1 的低字 ACCU1 - L
L   S5T#2S                    //将 16 位定时器常数装入累加器 1 的低字 ACCU1 - L
L   P#M5.6                    //将指向 M5.6 的指针装入累加器 1
```

**2. 直接寻址的装载与传送指令**

直接寻址在指令中直接给出存储器或寄存器的区域、长度和位置，例如用 MW200 指定位存储区中的字，地址为 200；下面是直接寻址的程序实例：

```
A   I0.0          //输入位 I0.0 的"与"(AND)操作
L   MB10          //将 8 位存储器字节装入累加器 1 最低的字节 ACCU1 - LL
L   DIW15         //将 16 位背景数据字装入累加器 1 的低字 ACCU1 - L
L   LD22          //将 32 位局域数据双字装入累加器 1
T   QB10          //将 ACCU1 - LL 中的数据传送到过程映像输出字节 QB10
T   MW14          //将 ACCU1 - L 中的数据传送到存储器字 MW14
T   DBD2          //将 ACCU1 中的数据传送到数据双字 DBD2
```

**3. 存储器间接寻址**

在存储器间接寻址指令中，给出一个作地址指针的存储器，该存储器的内容是操作数所在存储单元的地址。在循环程序中经常使用存储器间接寻址。

地址指针可以是字或双字，如定时器（T）、计数器（C）、数据块（DB）、功能块（FB）和功能（FC）的编号范围小于 65535，使用字指针。其他地址则要使用双字指针，如果要用双字格式的指针访问一个字、字节或双字存储器，必须保证指针的位编号为 0，例如 P#Q20.0。

```
L   QB[DBD 10]    //将输出字节装入累加器 1,输出字节的地址指针在数据双字 DBD10
                    中,如果 DBD10 的值为 2#0000 0000 0000 0000 0000 0000 0010 0000,
                    装入的是 QB4
A   M[LD 4]       //对存储器位作"与"运算,地址指针在数据双字 LD4 中,如果 LD4 的
值为 2#0000 0000 0000 0000 0000 0000 0010 0011,则是对 M4.3 进行操作
```

**【例 4-19】** 用循环指令和简洁寻址求从 MW60 开始存放的 5 个字的累加和。累加的结果用 MD50 保存，用临时局部变量 LD24 作为地址指针，LW32 作为循环次数计数器。

**解：**

```
        L     L#0               //将双整数 0 装入累加器 1
        T     MD    50          //当 I0.0 闭合时,RL0 = 1,跳转到 LAB1
        L     P#60.0
        T     LD    24          //地址指针
        L     5                 //将循环次数送入累加器 1
LAB1:   T     LW    32          //暂存循环计数值
        L     MW[LD  24]        //取数据
        ITD                     //整数转换成双整数
        L     MD    50          //取累加和
        +D                      //累加
        T     MD    50          //保存累加和
        L     LD    24          //取地址指针 1
        L     L#16
        +D    L#16
        +D                      //16 位即一个字,指针指向下一个字
```

```
T       LD      24      //保存地址指针值
T       LW      32      //循环计数器值装入累加器 1
LOOP    LAB1            //循环计数器值减 1,如非 0,跳转到 LAB1
NOP     0
```

每次累加完成后，为了使地址指针指向下一个字，地址指针的数值应加 16 或者加 P#2.0（1 个字由 16 位组成）。以上程序中的指令“L L#16”可改为“L P#2.0”，表示地址指针将加上一个字。如果是对字节操作，则指针应加 8。

**4. 寄存器间接寻址**

地址寄存器 AR1 和 AR2 的内容加上偏移量形成地址指针，指向数值所在的存储单元。其中第 0 ~ 2 位（xxx）为被寻址地址中位的编号（0 ~ 7），第 3 ~ 18 位为被寻址地址的字节编号（0 ~ 65535）。第 24 ~ 26 位（rrr）为被寻址地址的区域标识号，第 31 位 x = 0 为区域内的间接寻址，第 31 位 x = 1 为区域间的间接寻址。

第一种地址指针格式存储区的类型在指令中给出，例如 LDBB[AR1, P#6.0]。在某一存储区内寻址。第 24 ~ 26 位（rrr）应为 0。

第二种地址指针格式的第 24 ~ 26 位还包含存储区域标识符 rrr，区域间寄存器间接寻址。

如果要用寄存器指针访问一个字节、字或双字，必须保证指针中的位地址编号为 0。

指针常数#P5.0 对应的二进制数为 2#0000 0000 0000 0000 00000000 0010 1000。下面是区内间接寻址的例子：

```
L   P#5.0               //将间接寻址的指针装入累加器 1
LAR1                    //将累加器 1 中的内容送到地址寄存器 1
A   M[AR1, P#2.3]       //AR1 中的 P#5.0 加偏移量 P#2.3, 实际上是对 M7.3 进行操作
 =  Q[AR1, P#0.2]       //逻辑运算的结果送 Q5.2
L   DBW[AR1, P#18.0]    //将 DBW23 装入累加器 1
```

下面是区域间间接寻址的例子：

```
L   P#M6.0              //将存储器位 M6.0 的双字指针装入累加器 1
LAR1                    //将累加器 1 中的内容送到地址寄存器 1
TMW[AR1, P#50.0]        //将累加器 1 中的内容传送到存储器字 MW56
```

P#M6.0 对应的二进制数为 2#1000 0011 0000 0000 0000 0000 0011 0000。因为地址指针 P#M6.0 中已经包含有区域信息，使用间接寻址的指令 T W[AR1, P#50]中没有必要再用地址标识符 M。

**5. 装载时间值或计数值**

```
L    T5     //将定时器 T5 中的二进制时间值装入累加器 1 的低字中
LC   T5     //将定时器 T5 中的 BCD 码格式的时间值装入累加器 1 低字中
L    C3     //将计数器 C3 中的二进制计数值装入累加器 1 的低字中
LC   C16    //将计数器 C16 中的 BCD 码格式的值装入累加器 1 的低字中
```

**6. 地址寄存器的装载与传送指令**

可以不经过累加器 1，与地址寄存器 AR1 和 AR2 交换数据。下面是应用实例：

```
LAR1   DBD20      //将数据双字 DBD20 中的指针装入 AR1
LAR2   LD180      //将局域数据双字 LD180 中的指针装入 AR2
LAR1   P#M10.2    //将带存储区标识符的 32 位指针常数装入 AR1
LAR2   P#24.0     //将不带存储区标识符 32 位指针常数装入 AR2
TAR1   DBD20      //AR1 中的内容传送到数据双字 DBD20
TAR2   MD24       //AR2 中的内容传送到存储器双字 MD24
```

### 7. 装载与传送指令（MOVE）

对于初学者掌握指令表装载（L）与传送（T）是有些难度的，若读者先从梯形图中的传送指令学起，则容易理解，特别是对梯形图比较熟悉的读者更是如此。

当允许输入端的状态为“1”时，使能此指令，将 IN 端的数值输送到 OUT 端的目的地地址中，IN 和 OUT 有相同的信号状态，装载与传送指令（MOVE）的指令及参数见表 4-28。

表 4-28　装载与传送指令（MOVE）指令及参数

| LAD | 参　　数 | 数 据 类 型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| MOVE<br>—EN　ENO—<br>—IN　OUT— | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | OUT | 所有长度为 8、16 或 32 位的基本数据类型 | 目的地地址 | |
| | IN | | 源数据源 | |

用一个例子来说明装载与传送指令（MOVE）的使用，梯形图和指令表如图 4-74 所示，当 I0.0 闭合，MW20 中的数值（假设为 8），传送到目的地地址 MW22 中，结果是 MW20 和 MW22 中的数值都是 8。Q0.0 的状态与 I0.0 相同，也就是说，I0.0 闭合时，Q0.0 为“1”；I0.0 断开时，Q0.0 为“0”。

将图 4-74 所示的梯形图转化成指令表如下：

```
      A      I       0.0
      JNB    _001                  //如果 I0.0 =0,则跳转到标号_001 处
      L      MW      20            //MW20 的值装入累加器 1 的低字
      T      MW      22            //累加器 1 低字的内容传送到 MW22
      SET                          //将 RLO 置为 1
      SAVE                         //将 RLO 保存到 BR 位
      CLR                          //将 RLO 置为 0
_001：A      BR                    //状态字
      =      Q       0.0
```

**【例 4-20】**用传送指令，设计一个梯形图将存储区 MB0～MB3 的数据清除。

**解：**MB0～MB3 实际上就是 MD0，因此用一条传送指令即可，梯形图如图 4-75 所示。

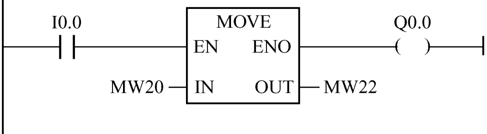



图 4-74　装载与传送梯形图指令示例

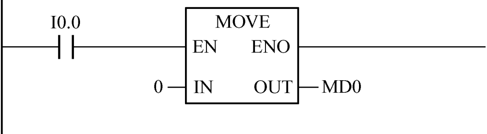



图 4-75　梯形图

**【关键点】**传送指令的输入端的数据类型可以是常数、字节、整数、双整数和实数，使用非常灵活。

**【例 4-21】**如图 4-76 所示的电气原理图，请编写电动机 Y－△起动控制程序。

**解：**前 10 s，Q0.0 和 Q0.1 线圈得电，星形起动，从第 10 s 只有 Q0.0 得电，从 11 s 开始，Q0.0 和 Q0.2 线圈得电，电动机为三角形运行。程序如图 4-77 所示。这种方法编写程序很简单，但浪费了宝贵的输出点资源。

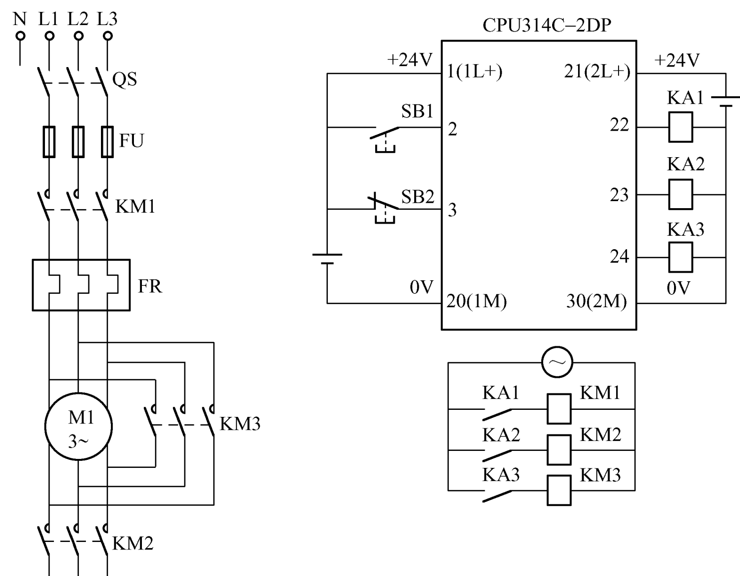
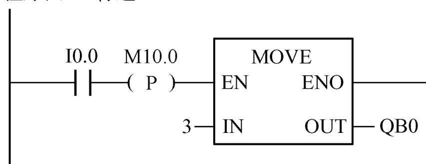
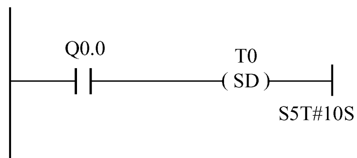
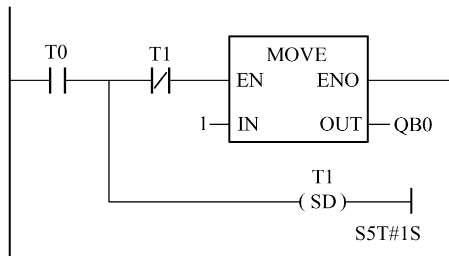
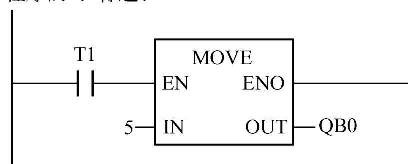



图 4-76　原理图



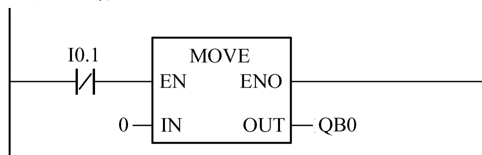




图 4-77　电动机 Y-△起动程序

以上程序是正确的，但需占用 8 个输出点，而真实使用的却只有 3 个，浪费了 5 个宝贵的输出点，因此从工程的角度考虑，不是一个“好”程序。改进的梯形图如图 4-78 所示，仍然采用以上方案，但只需要使用 3 个输出点，因此是一个“好”程序。

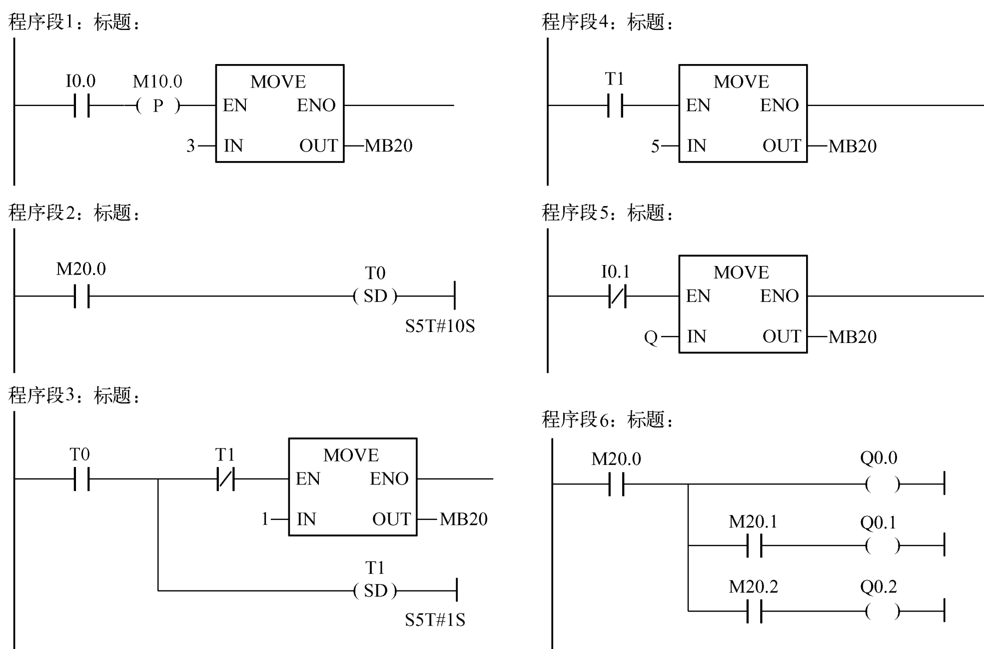



图 4-78　电动机 Y－△起动程序（改进后）

## 4.5.2　比较指令

STEP 7 提供了丰富的比较指令，可以满足用户的各种需要。STEP 7 中的比较指令可以对下列数据类型的数值进行比较。

① 两个整数的比较（每个整数为 16 位）。

② 两个双整数的比较（每个双整数为 32 位）。

③ 两个实数的比较（每个实数为 32 位）。

**【关键点】** 一个整数和一个双整数是不能直接进行比较的，因为它们之间的数据类型不同。一般先将整数转换成双整数，再对两个双整数进行比较。

比较指令有等于（EQ）、不等于（NQ）、大于（GT）、小于（LQ）、大于或等于（GE）和小于或等于（LE）。比较指令对输入 IN1 和 IN2 进行比较，如果比较结果为真，则逻辑运算结果 RLO 为“1”，反之则为“0”。

**1. 等于比较指令**

等于指令有整数等于比较指令、双整数等于比较指令和实数等于比较指令 3 种。整数等于比较指令和参数见表 4-29。

**表 4-29　整数等于比较指令和参数**

| LAD | 参　数 | 数 据 类 型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| CMP ==I<br>IN1<br>IN2 | IN1 | INT | 比较的第一个数值 | I、Q、M、D、L |
| | IN2 | INT | 比较的第二个数值 | |

用一个例子来说明整数等于比较指令，梯形图和指令表如图 4-79 所示。当 I0.0 闭合时，激活比较指令，MW0 中的整数和 MW2 中的整数比较，若两者相等，则 Q0.0 输出为“1”，若两者不相等，则 Q0.0 输出为“0”。在 I0.0 不闭合时，Q0.0 的输出为“0”。IN1 和 IN2 可以为常数。

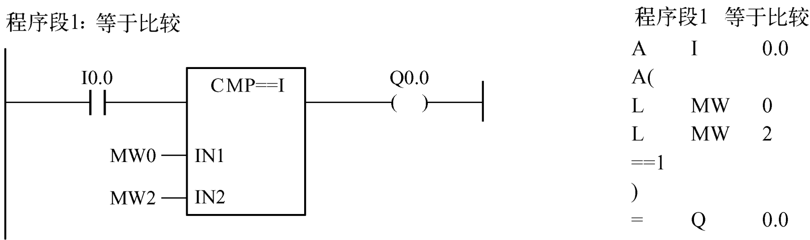



图 4-79　整数等于比较指令示例

双整数等于比较指令和实数等于比较指令的使用方法与整数等于比较指令类似，只不过 IN1 和 IN2 的参数类型分别为双整数和实数。

**2. 不等于比较指令**

不等于比较指令有整数不等于比较指令、双整数不等于比较指令和实数不等于比较指令 3 种。整数不等于比较指令和参数见表 4-30。

**表 4-30　整数不等于比较指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| CMP <>I<br>IN1<br>IN2 | IN1 | INT | 比较的第一个数值 | I、Q、M、D、L |
| | IN2 | INT | 比较的第二个数值 | |

用一个例子来说明整数不等于比较指令，梯形图和指令表如图 4-80 所示。当 I0.0 闭合时，激活比较指令，MW0 中的整数和 MW2 中的整数比较，若两者不相等，则 Q0.0 输出为“1”，若两者相等，则 Q0.0 输出为“0”。在 I0.0 不闭合时，Q0.0 的输出为“0”。IN1 和 IN2 可以为常数。

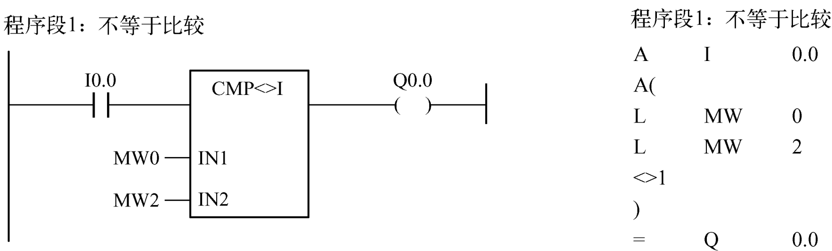



图 4-80　整数不等于比较指令示例

双整数不等于比较指令和实数不等于比较指令的使用方法与整数不等于比较指令类似，只不过 IN1 和 IN2 的参数类型分别为双整数和实数。使用比较指令的前提是数据类型必须相同。

**3. 小于比较指令**

小于比较指令有整数小于比较指令、双整数小于比较指令和实数小于比较指令 3 种。双整数小于比较指令和参数见表 4-31。

**表 4-31　双整数小于比较指令和参数**

| LAD | 参　　数 | 数 据 类 型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| CMP <D<br>IN1<br>IN2 | IN1 | DINT | 比较的第一个数值 | I、Q、M、D、L |
| | IN2 | DINT | 比较的第二个数值 | |

用一个例子来说明双整数小于比较指令，梯形图和指令表如图 4-81 所示。当 I0. 0 闭合时，激活双整数小于比较指令，MD0 中的双整数和 MD4 中的双整数比较，若前者小于后者，则 Q0. 0 输出为“1”，否则 Q0. 0 输出为“0”。在 I0. 0 不闭合时，Q0. 0 的输出为“0”。IN1 和 IN2 可以为常数。

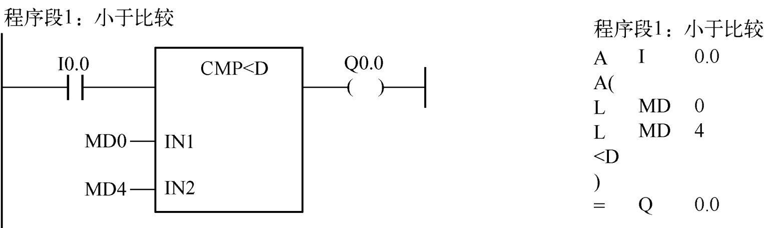



图 4-81　双整数小于比较指令示例

整数小于比较指令和实数小于比较指令的使用方法与双整数小于比较指令类似，只不过 IN1 和 IN2 的参数类型分别为整数和实数。使用比较指令的前提是数据类型必须相同。

**4. 大于等于比较指令**

大于等于比较指令有整数大于等于比较指令、双整数大于等于比较指令和实数大于等于比较指令 3 种。实数大于等于比较指令和参数见表 4-32。

**表 4-32　实数大于等于比较指令和参数**

| LAD | 参　　数 | 数 据 类 型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| CMP>=R<br>IN1<br>IN2 | IN1 | REAL | 比较的第一个数值 | I、Q、M、D、L |
| | IN2 | REAL | 比较的第二个数值 | |

用一个例子来说明实数大于等于比较指令，梯形图和指令表如图 4-82 所示。当 I0. 0 闭合时，激活比较指令。MD0 中的实数和 MD4 中的实数比较，若前者大于或者等于后者，则 Q0. 0 输出为“1”，否则 Q0. 0 输出为“0”。在 I0. 0 不闭合时，Q0. 0 的输出为“0”。IN1 和 IN2 可以为常数。

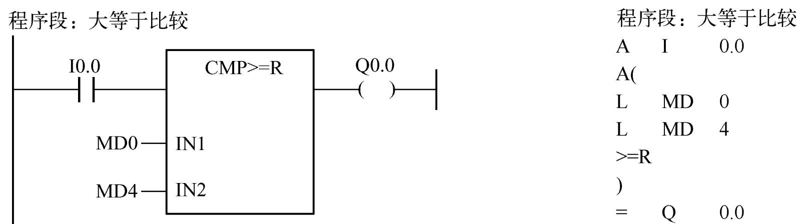



图 4-82　实数大于等于比较指令示例

整数大于等于比较指令和双整数大于等于比较指令的使用方法与实数大于等于比较指令类似，只不过 IN1 和 IN2 的参数类型分别为整数和双整数。使用比较指令的前提是数据类型必须相同。

小于等于比较指令和小于比较指令类似，大于比较指令和大于等于比较指令类似，在此不再讲述小于等于比较指令和大于比较指令。

**【例 4-22】** 某设备上的控制器是 CPU 314C-2DP，设备上有一个光电传感器，检测工件，每检测到 1 只工件，计数一次，当计数到 3 只时，CPU 发出一个信号装箱，请设计梯形图。

**解：** 梯形图如图 4-83 所示。光电传感器每检测一个工件时，计数器 C0 计 1 次数，当前计数值存放在 MW12 中，当计数 3 次时（MW12 中的数值大于等于 3），发出装箱信号 Q0.0，与此同时定时器 T0 开始定时，2 s 后对计数器 C0 复位，重新计数。

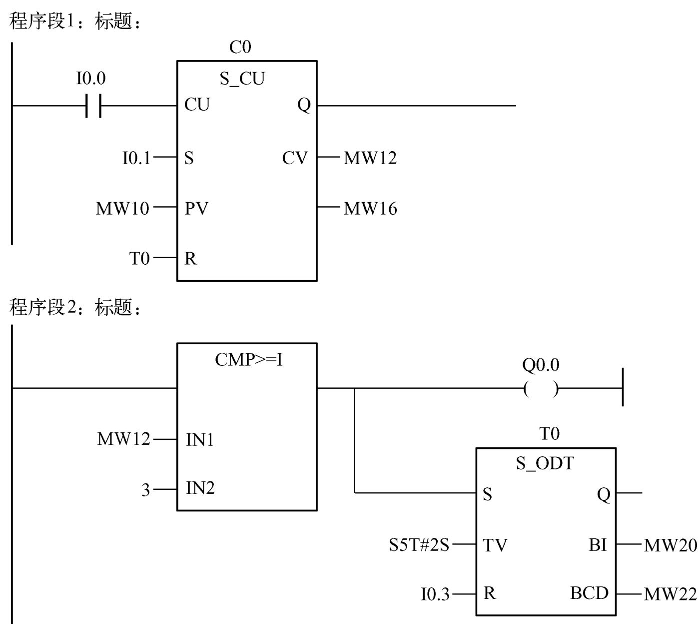



图 4-83　梯形图

## 4.5.3　转换指令

转换指令是将一种数据格式转换成另外一种格式进行存储。例如，要让一个整型数据和双整型数据进行算术运算，一般要将整型数据转换成双整型数据。

STEP 7 的转换指令见表 4-33。

**表 4-33　转换指令**

| STL | LAD | 说　明 |
|---|---|---|
| BTI | BCD_I | 将累加器 1 中的 3 位 BCD 码转换成整数 |
| ITB | I_BCD | 将累加器 1 中的整数转换成 3 位 BCD 码 |
| BTD | BCD_DI | 将累加器 1 中的 7 位 BCD 码转换成双整数 |
| DTB | DI_BCD | 将累加器 1 中的双整数转换成 7 位 BCD 码 |
| DTR | DI_R | 将累加器 1 中的双整数转换成浮点数 |
| ITD | I_DI | 将累加器 1 中的整数转换成双整数 |
| RND | ROUND | 将浮点数转换为四舍五入的双整数 |
| RND + | CEIL | 将浮点数转换为大于等于它的最小双整数 |
| RND - | FLOOR | 将浮点数转换为小于等于它的最大双整数 |
| TRUNC | TRUNC | 将浮点数转换为截位取整的双整数 |
| CAW | - | 交换累加器 1 低字中两个字节的位置 |
| CAD | - | 交换累加器 1 中 4 个字节的顺序 |

**1. BCD 转换成整数（BTI）**

（1）BCD 码的格式

BCD 码是比较有用的，3 位格式如图 4-84 所示，二进制的 0 ~ 3 位是个位，4 ~ 7 位是十位，8 ~ 11 位是百位，12 ~ 15 位是符号位。7 位格式如图 4-85 所示，二进制的 0 ~ 3 位是个位，4 ~ 7 位是十位，8 ~ 11 位是百位，12 ~ 15 位是千位，16 ~ 19 位是万位，20 ~ 23 位是十万位，24 ~ 27 位是百万位，28 ~ 31 位是符号位。

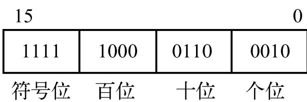



图 4-84　3 位 BCD 码的格式

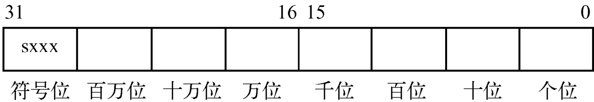



图 4-85　7 位 BCD 码的格式

（2）BCD 转换成整数指令（BTI）

BCD 转换成整数指令是将 IN 指定的内容以 BCD 码二 ~ 十进制格式读出，并将其转换为整数格式，输出到 OUT 端。如果 IN 端指定的内容超出 BCD 码的范围（例如 4 位二进制数出现 1010 ~ 1111 的几种组合），则执行指令时将会发生错误，使 CPU 进入 STOP 方式。BCD 转换成整数指令和参数见表 4-34。

**表 4-34　BCD 转换成整数指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存储区 |
|---|---|---|---|---|
| BCD_I（EN ENO / IN OUT） | EN | BOOL | 使能（允许输入） | I、Q、M、D、L |
| | IN | WORD | 输入的 BCD 数 | |
| | ENO | BOOL | 允许输出 | |
| | OUT | INT | BCD 的整数 | |

用一个例子来说明 BCD 转换成整数指令，梯形图和指令表如图 4-86 所示。当 I0.0 闭合时，激活 BCD 转换成整数指令，IN 中的 BCD 数用十六进制表示为 16#22（就是十进制的 22），转换完成后 OUT 端的 MW0 中的整数的十六进制是 16#16。

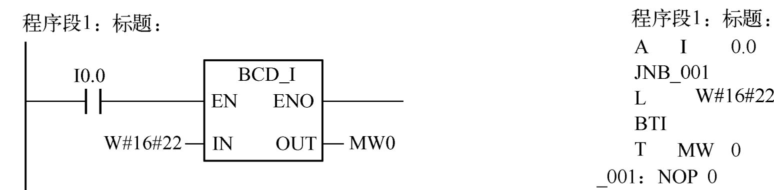



图 4-86 BCD 转换成整数指令示例

**2. 整数转换成 BCD（ITB）**

整数转换成 BCD 指令是将 IN 端指定的内容以整数的格式读入，然后将其转换为 BCD 码格式输出到 OUT 端。如果 IN 端的整数大于 999，PLC 不停机，仍然正常运行。由于字的 BCD 码最大只能表示 C#999（最高 4 位为符号位）。若 IN 端的内容大于 999，CPU 将 IN 端的内容直接送到 OUT 端输出，不经过 I_BCD 的转换。这时 OUT 输出的内容可能超出 BCD 码的范围。另外 OUT 端的内容若为 BCD 码，也有可能是超过 999 的整数转换出来的，例如整数 2457 通过 I_BCD 指令以后，OUT 的值为 C#999。因此在使用 I_BCD 指令时应该保证整数小于等于 999。此外，如果 IN 端的整数为负整数时。转换出的 BCD 码最高 4 位为“1”。整数转换成 BCD 指令和参数见表 4-35。

**表 4-35 整数转换成 BCD 指令和参数**

| LAD | 参 数 | 数 据 类 型 | 说 明 | 存 储 区 |
|---|---|---|---|---|
| I_BCD<br>EN ENO<br>IN OUT | EN | BOOL | 使能（允许输入） | I、Q、M、D、L |
| | IN | INT | 输入的整数 | |
| | ENO | BOOL | 允许输出 | |
| | OUT | WORD | 整数转化成的 BCD 数 | |

用一个例子来说明整数转换成 BCD 指令，梯形图和指令表如图 4-87 所示。当 I0.0 闭合时，激活整数转换成 BCD 指令，IN 中的整数存储在 MW0 中，（假设用十六进制表示为 16 #16），转换完成后 OUT 端的 MW2 中的 BCD 数是 16#22。

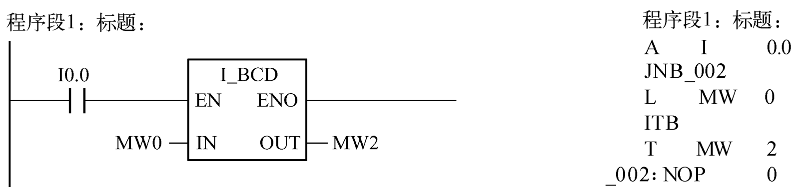



图 4-87 整数转换成 BCD 指令示例

**3. 整数转换成双整数（ITD）**

整数转换成双整数指令是将 IN 端指定的内容以整数的格式读入，然后将其转换为双整数码格式输出到 OUT 端。整数转换成双整数指令和参数见表 4-36。

**表 4-36　整数转换成双整数指令和参数**

| LAD | 参　　数 | 数 据 类 型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| I_DI<br>EN　ENO<br>IN　OUT | EN | BOOL | 使能（允许输入） | I、Q、M、D、L |
| | IN | INT | 输入的整数 | |
| | ENO | BOOL | 允许输出 | |
| | OUT | DINT | 整数转化成的 BCD 数 | |

用一个例子来说明整数转换成双整数指令，梯形图和指令表如图 4-88 所示。当 I0.0 闭合时，激活整数转换成双整数指令，IN 中的整数存储在 MW0 中，（假设用十六进制表示为 16#0016），转换完成后 OUT 端的 MD0 中的双整数是 16#0000 0016。转换前后的示意如图 4-89 所示。

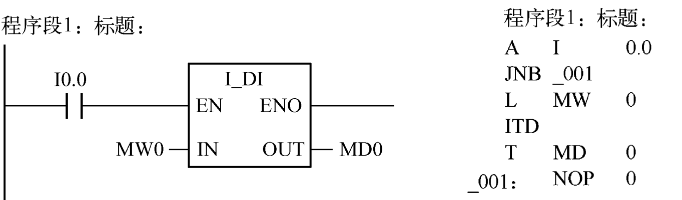



图 4-88　整数转换成双整数指令示例

整数转换成双整数 MW0 = MD0，其大小并未发生改变。但从图 4-64 可以看出数据的变化：转换之前，MB1 = 16#16，而转换之后，MB1 = 16#00，MB3 = 16#16。数据大小虽未改变，但数据的存放位置发生了改变，这点初学者容易忽略。

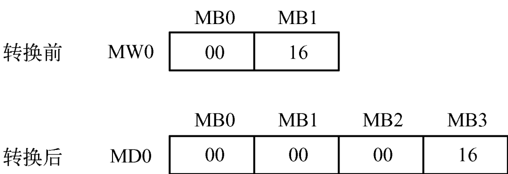



图 4-89　整数转换成双整数前后的示意图

**4. 双整数转换成实数（DTR）**

双整数转换成实数指令是将 IN 端指定的内容以双整数的格式读入，然后将其转换为实数码格式输出到 OUT 端。实数格式在后续算术计算中是很常用的，如 3.14 就是实数形式。双整数转换成实数指令和参数见表 4-37。

**表 4-37　双整数转换成实数指令和参数**

| LAD | 参　　数 | 数 据 类 型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| DI_R<br>EN　ENO<br>IN　OUT | EN | BOOL | 使能（允许输入） | I、Q、M、D、L |
| | IN | DINT | 输入的双整数 | |
| | ENO | BOOL | 允许输出 | |
| | OUT | REAL | 双整数转化成的实数 | |

用一个例子来说明双整数转换成实数指令，梯形图和指令表如图 4-90 所示。当 I0.0 闭合时，激活双整数转换成实数指令，IN 中的双整数存储在 MD0 中，（假设用十进制表示为 L#16），转换完成后 OUT 端的 MD4 中的实数是 16.0。一个实数要用 4 个字节存储。

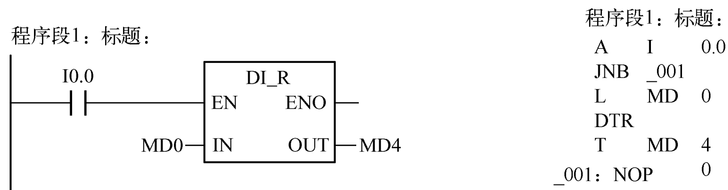



图 4-90　双整数转换成实数指令示例

**【例 4-23】** 请设计梯形图程序，将整数 16#22 转化成实数，并保存在 MD10 中。

**解：** STEP 7 中没有将整数直接转化成实数的指令，但可以通过数次转换将整数转换成实数，先将整数转换成双整数，再将双整数转换成实数。梯形图如图 4-91 所示。

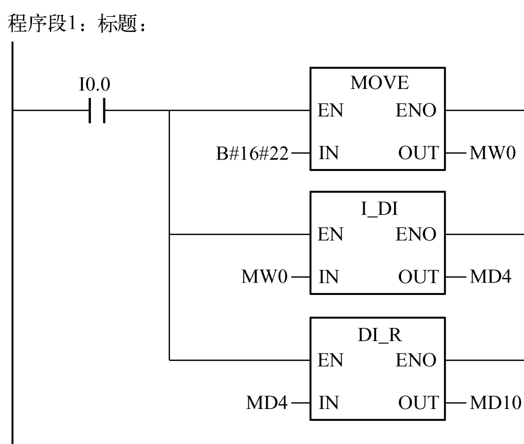



图 4-91　梯形图

### 5. 实数四舍五入为双整数（ROUND）

ROUND 指令是将实数进行四舍五入取整后转换成双整数的格式。实数四舍五入为双整数指令和参数见表 4-38。

**表 4-38　实数四舍五入为双整数指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| ROUND（EN ENO / IN OUT） | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN | REAL | 实数（浮点型） | |
| | OUT | DINT | 四舍五入后为双整数 | |

用一个例子来说明实数四舍五入为双整数指令，梯形图和指令表如图 4-92 所示。当 I0.0 闭合时，激活实数四舍五入指令，IN 中的实数存储在 MD0 中。假设这个实数为 3.14，进行四舍五入运算后 OUT 端的 MD4 中的双整数是 L#3；假设这个实数为 3.88，进行四舍五入运算后 OUT 端的 MD4 中的双整数是 L#4。

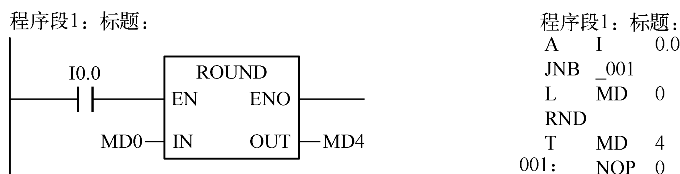



图 4-92　实数四舍五入为双整数指令示例

### 4.5.4　移位与循环指令

STEP 7 移位指令能将累加器的内容逐位向左或者向右移动。移动的位数由 N 决定。向左移 N 位相当于累加器的内容乘以 $2^N$，向右移相当于累加器的内容除以 $2^N$。移位指令在逻辑控制中使用也很方便。移位与循环指令见表 4-39。

**表 4-39　移位与循环指令**

| 名　　称 | 语　句　表 | 梯　形　图 | 描　　述 |
|---|---|---|---|
| 有符号整数右移 | SSI | SHR_I | 整数逐位右移，空出的位添上符号位 |
| 有符号双整数右移 | SSD | SHR_DI | 双整数逐位右移，空出的位添上符号位 |
| 16 位字左移 | SLW | SHL_W | 字逐位左移，空出的位添 0 |
| 16 位字右移 | SRW | SHR_W | 字逐位右移，空出的位添 0 |
| 16 位双字左移 | SLD | SHL_DW | 双字逐位左移，空出的位添 0 |
| 16 位双字右移 | SRD | SHR_DW | 双字逐位右移，空出的位添 0 |
| 双字循环左移 | RLD | ROL_DW | 双字循环左移 |
| 双字循环右移 | RRD | ROR_DW | 双字循环右移 |

**1. 字左移（SHL_W）**

当字左移（SHL_W）指令的 EN 位为高电平“1”时，将执行移位指令，将 IN 端指定的内容送入累加器 1 低字中，并左移 N 端指定的位数，然后写入 OUT 端指令的目的地址中。字左移（SHL_W）指令和参数见表 4-40。

**表 4-40　字左移（SHL_W）指令和参数**

| LAD | 参　　数 | 数 据 类 型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| SHL_W<br>EN　ENO<br>IN　OUT<br>N | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN | WORD | 移位对象 | |
| | N | WORD | 移动的位数 | |
| | OUT | WORD | 移动操作的结果 | |

用一个例子来说明字左移指令，梯形图和指令表如图 4-93 所示。当 I0.0 闭合时，激活左移指令，IN 中的字存储在 MW0 中，假设这个数为 2#1001 1101 1111 1011，向左移 4 位后，OUT 端的 MW0 中的数是 2#1101 1111 1011 0000，字左移指令示意图如图 4-94 所示。

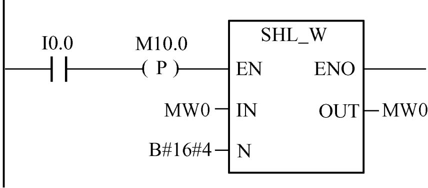
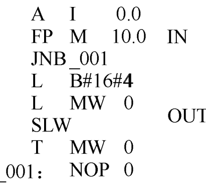



图 4-93　字左移指令示例



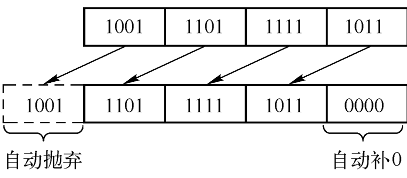
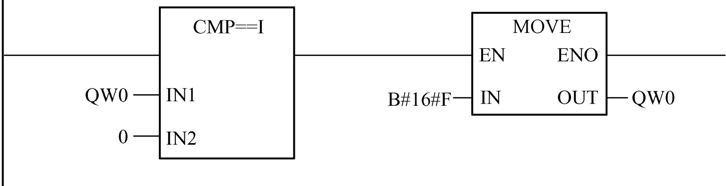




图 4-94　字左移指令示意图

【**关键点**】图 4-93 中的程序有一个上升沿，这样 I0.0 每闭合一次，左移 4 位，若没有上升沿，那么闭合一次，可能左移很多次。这点读者要特别注意。

【**例 4-24**】有 16 盏灯，上电时，1 ~ 4 盏亮，1 s 后 5 ~ 8 盏亮，1 ~ 4 盏灭，如此不断循环，请编写程序。

**解：**M0.5 是设定为 1 s 的脉冲信号，梯形图如图 4-95 所示。可以看出，用移位指令编写程序，很简洁。



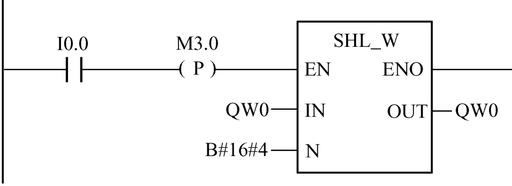




图 4-95　梯形图

**2. 字右移（SHR_W）**

当字右移（SHR_W）指令的 EN 位为高电平“1”时，将执行移位指令，将 IN 端指令的内容送入累加器 1 低字中，并右移 N 端指定的位数，然后写入 OUT 端指令的目的地址中。字右移（SHR_W）指令和参数见表 4-41。

**表 4-41　字右移（SHR_W）指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存储区 |
|---|---|---|---|---|
| SHR_W（EN、ENO、IN、OUT、N） | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN | WORD | 移位对象 | |
| | N | WORD | 移动的位数 | |
| | OUT | WORD | 移动操作的结果 | |

用一个例子来说明字右移指令，梯形图和指令表如图 4-96 所示。当 I0.0 闭合时，激活右移指令，IN 中的字存储在 MW0 中，假设这个数为 2#1001 1101 1111 1011，向右移 4 位后，OUT 端的 MW0 中的数是 2#0000 1001 1101 1111，字右移指令示意图如图 4-97 所示。

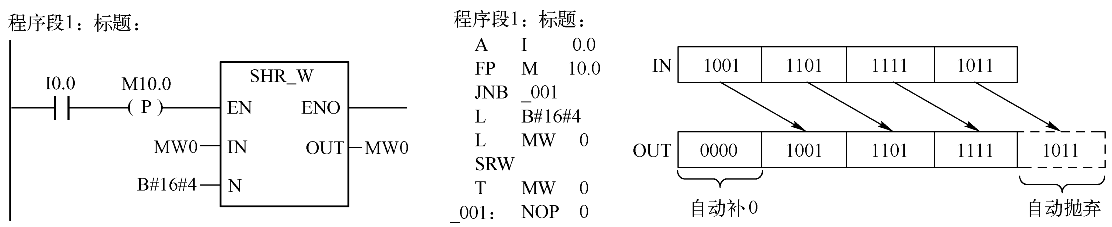



图 4-96　字右移指令示例　　　图 4-97　字右移指令示意图

**3. 双字左移（SHL_DW）**

当双字左移（SHL_DW）指令的 EN 位为高电平“1”时，将执行移位指令，将 IN 端指定的内容左移 N 端指定的位数，然后写入 OUT 端指令的目的地址中。双字左移（SHL_DW）指令和参数见表 4-42。

**表 4-42　双字左移指令（SHL_DW）和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| SHL_DW（EN ENO / IN OUT / N） | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN | DWORD | 移位对象 | |
| | N | WORD | 移动的位数 | |
| | OUT | DWORD | 移动操作的结果 | |

用一个例子来说明双字左移指令，梯形图和指令表如图 4-98 所示。当 I0.0 闭合时，激活双字左移指令，IN 中的双字存储在 MD0 中，假设这个数为 16#87654321，向左移 4 位后（半个字节），OUT 端的 MD0 中的数是 16#76543210。

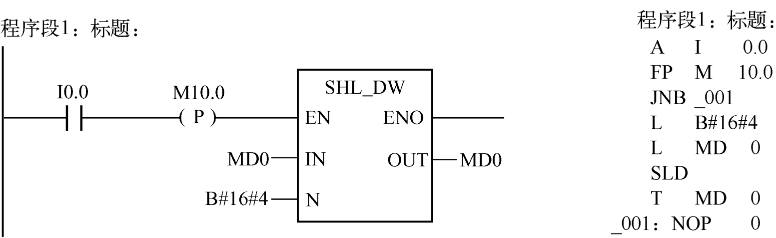



图 4-98　双字左移指令示例

**4. 双字右移（SHR_DW）**

当双字右移（SHR_DW）指令的 EN 位为高电平“1”时，将执行移位指令，将 IN 端指定的内容右移 N 端指定的位数，然后写入 OUT 端指令的目的地址中。双字右移（SHR_DW）指令和参数见表 4-43。

表 4-43　双字右移（SHR_DW）指令和参数

| LAD | 参　　数 | 数据类型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| SHR_DW<br>EN　ENO<br>IN　OUT<br>N | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN | DWORD | 移位对象 | |
| | N | WORD | 移动的位数 | |
| | OUT | DWORD | 移动操作的结果 | |

用一个例子来说明双字右移指令，梯形图和指令表如图 4-99 所示。当 I0.0 闭合时，激活双字右移指令，IN 中的双字存储在 MD0 中，假设这个数为 16#12345678，向左移 4 位后，OUT 端的 MD0 中的数是 16#01234567。

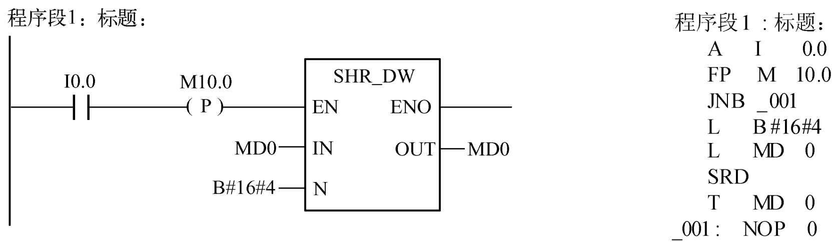



图 4-99　双字右移指令示例

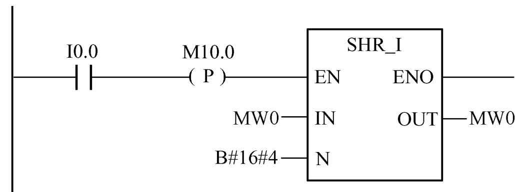
**5. 整数右移（SHR_I）**

当整数右移（SHR_I）指令的 EN 位为高电平“1”时，将执行移位指令，将 IN 端指定的内容右移 N 端指定的位数，然后写入 OUT 端指令的目的地址中，与字的右移不同的是整数移位时，按照低位丢失、高位补符号位状态的原则，即正数高位补“0”，而负数补“1”。整数右移（SHR_I）指令和参数见表 4-44。

表 4-44　整数右移指令（SHR_I）和参数

| LAD | 参　　数 | 数据类型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| SHR_I<br>EN　ENO<br>IN　OUT<br>N | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN | WORD | 移位对象 | |
| | N | WORD | 移动的位数 | |
| | OUT | WORD | 移动操作的结果 | |

用一个例子来说明整数右移指令，梯形图和指令表如图 4-100 所示。当 I0.0 闭合时，激活整数右移指令，IN 中的整数存储在 MW0 中，假设这个数为 2#0001 1101 1111 1011，向右移 4 位后，OUT 端的 MW0 中的数是 2#0000 0001 1101 1111，而假设这个数为 2#1001 1101 1111 1011，向右移 4 位后，OUT 端的 MW0 中的数是 2#1111 1001 1101 1111，其示意图如图 4-101 所示。




```
程序段1：标题：
  A   I    0.0
  FP  M   10.0
  JNB _001
  L   B#16#4
  L   MW  0
  SSI
  T   MW  0
_001： NOP 0
```

图 4-100　整数右移指令示例

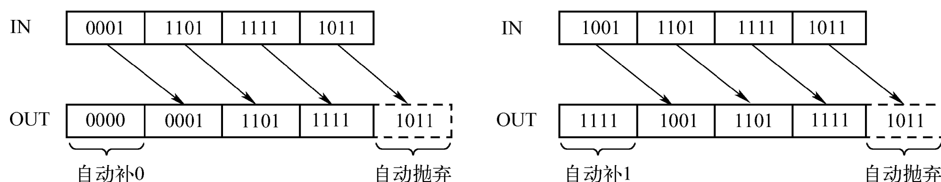



图 4-101　整数右移指令示意图

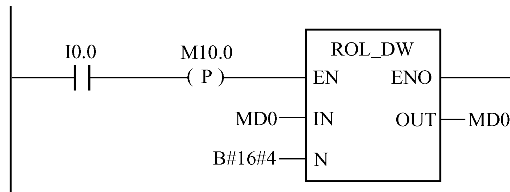
**6. 双字循环左移（ROL_DW）**

当双字循环左移（ROL_DW）指令的 EN 位为高电平“1”时，将执行双字循环左位指令，将 IN 端指定的内容循环左移 N 端指定的位数，然后写入 OUT 端指令的目的地址中。双字循环左移（ROL_DW）指令和参数见表 4-45。

**表 4-45　双字循环左移（ROL_DW）指令和参数**

| LAD | 参　数 | 数 据 类 型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| ROL_DW: EN ENO / IN OUT / N | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN | DWORD | 移位对象 | |
| | N | WORD | 移动的位数 | |
| | OUT | DWORD | 移动操作的结果 | |

用一个例子来说明双字循环左移（ROL_DW）指令的应用，梯形图和指令表如图 4-102 所示。当 I0.0 闭合时，激活双字循环左移指令，IN 中的双字存储在 MD0 中，假设这个数为 2#1001 1101 1111 1011 1001 1101 1111 1011，除最高 4 位外，其余各位向左移 4 位后，双字的最高 4 位，循环到双字的最低 4 位，结果是 OUT 端的 MD0 中的数是 2#1101 1111 1011 1001 1101 1111 1011 1001，其示意图如图 4-103 所示。




```
程序段1：标题：
  A     I       0.0
  FP    M      10.0
  JNB   _001
  L     B#16#4
  L     MD      0
  RLD
  T     MD      0
_001：  NOP     0
```

图 4-102　双字循环左移指令示例

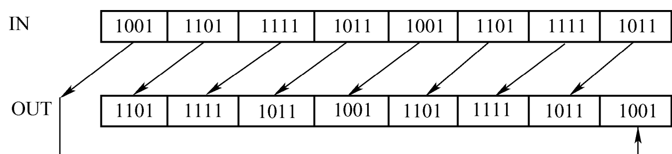


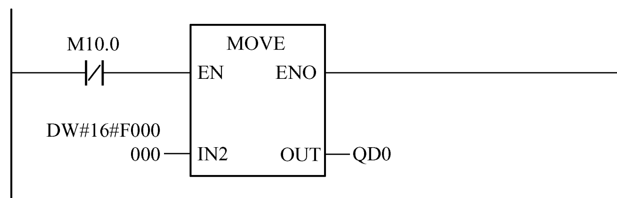

图 4-103　双字循环左移指令示意图

**【例 4-25】** 有 32 盏灯，上电时，1～4 盏亮，1 s 后 5～8 盏亮，1～4 盏灭，如此不断循环，请编写程序。

**解：** M0.5 是设定的 1 s 脉冲信号，梯形图如图 4-104 所示。可以看出，用循环指令编写程序，很简洁。此题还有多种解法，请读者自己思考。

程序段1：标题：

首次扫描时，Q0.0、Q0.1、Q0.2、Q0.3亮；




程序段2：标题：

注释：




程序段3：标题：

每秒，移4位，且不断循环；

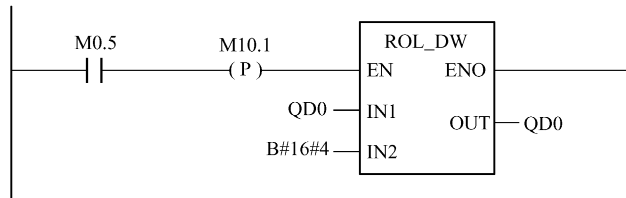



图 4-104　梯形图

### 7. 双字循环右移（ROR_DW）

当双字循环右移（ROR_DW）指令的 EN 位为高电平“1”时，将执行双字循环右移指令，将 IN 端指定的内容循环右移 N 端指定的位数，然后写入 OUT 端指令的目的地址中。双字循环右移（ROR_DW）指令和参数见表 4-46。

**表 4-46　双字循环右移（ROR_DW）指令和参数**

| LAD | 参　　数 | 数据类型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| ROR_DW<br>EN　ENO<br>IN1<br>IN2　OUT | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN | DWORD | 移位对象 | |
| | N | WORD | 移动的位数 | |
| | OUT | DWORD | 移动操作的结果 | |

用一个例子来说明双字循环右移（ROR_DW）指令的应用，梯形图和指令表如图 4-105 所示。当 I0.0 闭合时，激活双字循环右移指令，IN 中的双字存储在 MD0 中，假设这个数为 2#1001 1101 1111 1011 1001 1101 1111 1011，除最低 4 位外，其余各位向右移 4 位后，双字的最低 4 位，循环到双字的最高 4 位，结果是 OUT 端的 MD0 中的数是 2#1011 1001 1101 1111 1011 1001 1101 1111，其示意图如图 4-106 所示。

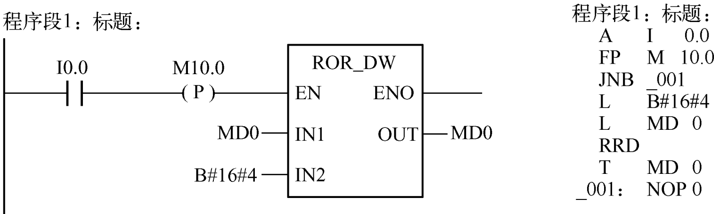



图 4-105　双字循环右移指令示例

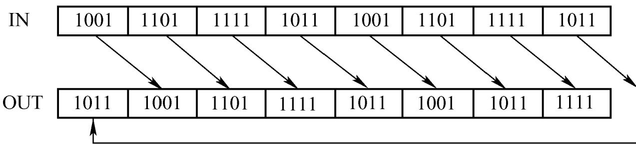



图 4-106　双字循环右移指令示意图

### 4.5.5　算术运算指令

算术运算指令非常重要，在模拟量的处理、PID 控制等很多场合都要用到算术运算。算术运算又分为整数算术运算和浮点数算术运算。

**1. 整数算术运算**

整数算术运算又分为加法运算、减法运算、乘法运算和除法运算，其中每种运算方式又有整数型和双整数型两种。

（1）整数加（ADD_I）

当允许输入端 EN 为高电平“1”时，输入端 IN1 和 IN2 中的整数相加，结果送入 OUT 中。如果该结果超出了整数（16 位）允许的范围，OV 位和 OS 位将为“1”并且 ENO 为逻辑“0”，这样便不执行此数学框后 ENO 连接的其他函数。IN1 和 IN2 中的数可以是常数。整数加的表达式是：IN1 + IN2 = OUT。

整数加（ADD_I）指令和参数见表 4-47。

**表 4-47　整数加（ADD_I）指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存储区 |
|---|---|---|---|---|
| ADD_I<br>EN ENO<br>IN1<br>IN2 OUT | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN1 | INT | 相加的第 1 个值 | |
| | IN2 | INT | 相加的第 2 个值 | |
| | OUT | INT | 相加的结果 | |

用一个例子来说明整数加（ADD_I）指令，梯形图和指令表如图 4-107 所示。当 I0.0 闭合时，激活整数加指令，IN1 中的整数存储在 MW0 中，假设这个数为 11，IN2 中的整数存储在 MW2 中，假设这个数为 21，整数相加的结果存储在 OUT 端的 MW4 中的数是 32。由于没有超出计算范围，所以 Q0.0 输出为“1”。假设 IN1 中的整数为 9999，IN2 中的整数为 30000，整数相加的结果存储在 OUT 端的 MW4 中的数是 -25537。由于超出计算范围，所以 Q0.0 输出为“0”。

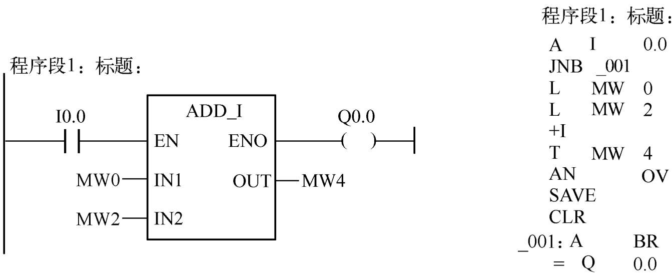



图 4-107　整数加（ADD_I）指令示例

**【例 4-26】** 有一个电炉，加热功率有 1000 W、2000 W 和 3000 W 三个档次，电炉有 1000 W 和 2000 W 两种电加热丝。要求用一个按钮选择三个加热档，当按一次按钮时，1000 W 电阻丝加热，即第一档；当按两次按钮时，2000 W 电阻丝加热，即第二档；当按三次按钮时，1000 W 和 2000 W 电阻丝同时加热，即第三档；当按四次按钮时停止加热，请编写程序。

**解：** 梯形图如图 4-108 所示。

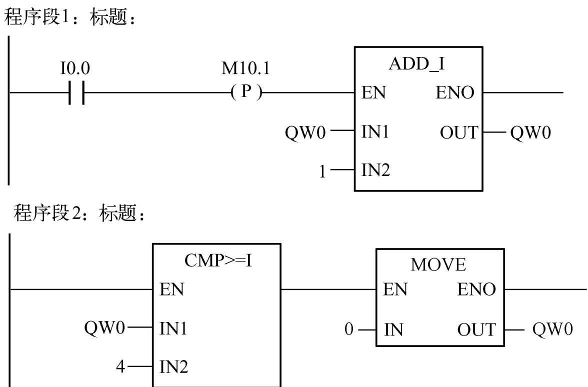



图 4-108　梯形图

如图 4-108 所示的梯形图程序，没有逻辑错误，但实际上有两处缺陷，一是上电时没有对 Q0.0 ~ Q0.2 复位，二是浪费了多达 13 个输出点，这在实际工程应用中是不允许的。对以上程序进行改进，如图 4-109 所示。

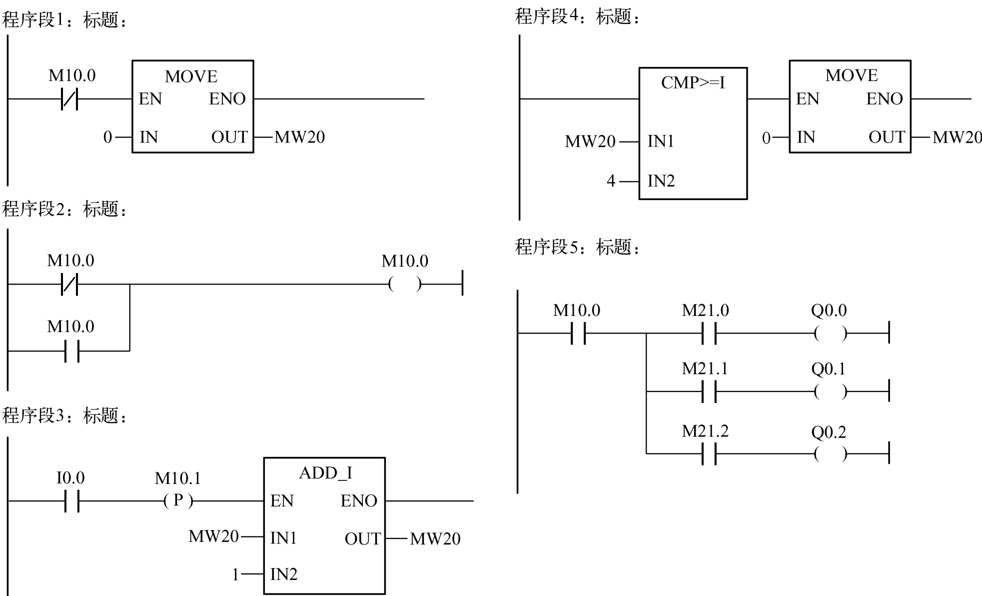


图 4-109　梯形图（改进后）

双整数加（ADD_DI）指令与整数加（ADD_I）指令类似，只不过其数据类型为双整数，在此不赘述。

（2）双整数减（SUB_DI）

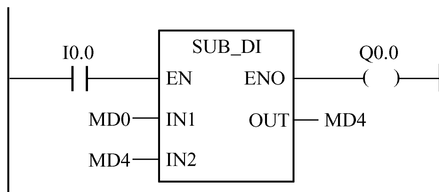
当允许输入端 EN 为高电平“1”时，输入端 IN1 和 IN2 中的双整数相减，结果送入 OUT 中。如果该结果超出了双整数（32 位）允许的范围，OV 位和 OS 位将为“1”并且 ENO 为逻辑“0”，这样便不执行此数学框后 ENO 连接的其他函数。IN1 和 IN2 中的数可以是常数。双整数减的表达式是：IN1 - IN2 = OUT。

双整数减（SUB_DI）指令和参数见表 4-48。

**表 4-48　双整数减（SUB_DI）指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| SUB_DI<br>EN　ENO<br>IN1<br>IN2　OUT | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN1 | DINT | 被减数 | |
| | IN2 | DINT | 减数 | |
| | OUT | DINT | 差 | |

用一个例子来说明双整数减（SUB_DI）指令，梯形图和指令表如图 4-110 所示。当 I0.0 闭合时，激活双整数减指令，IN1 中的双整数存储在 MD0 中，假设这个数为 L#22，IN2 中的双整数存储在 MD4 中，假设这个数为 L#11，双整数相减的结果存储在 OUT 端的 MD4 中的数是 L#11。由于没有超出计算范围，所以 Q0.0 输出为“1”。




程序段1：标题：

```
        A     I     0.0
        JNB   _001
        L     MD    0
        L     MD    4
        -D
        T     MD    4
        AN    OV
        SAVE
        CLR
_001:   A     BR
        =     Q     0.0
```

图 4-110　双整数减（SUB_DI）指令示例

整数减（SUB_I）指令与双整数减（SUB_DI）指令类似，只不过其数据类型为整数，在此不赘述。

（3）整数乘（MUL_I）

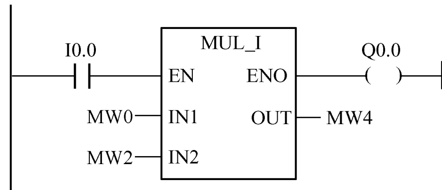
当允许输入端 EN 为高电平“1”时，输入端 IN1 和 IN2 中的整数相乘，结果送入 OUT 中。如果该结果超出了整数允许的范围，OV 位和 OS 位将为“1”并且 ENO 为逻辑“0”，这样便不执行此数学框后 ENO 连接的其他函数。IN1 和 IN2 中的数可以是常数。整数乘的表达式是：IN1 × IN2 = OUT。

整数乘（MUL_I）指令和参数见表 4-49。

**表 4-49　整数乘（MUL_I）指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| MUL_I（EN、ENO、IN1、IN2、OUT） | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN1 | INT | 相乘的第 1 个值 | |
| | IN2 | INT | 相乘的第 2 个值 | |
| | OUT | INT | 相乘的结果（积） | |

用一个例子来说明整数乘（MUL_I）指令，梯形图和指令表如图 4-111 所示。当 I0.0 闭合时，激活整数乘指令，IN1 中的整数存储在 MW0 中，假设这个数为 11，IN2 中的整数存储在 MW2 中，假设这个数为 11，整数相乘的结果存储在 OUT 端的 MW4 中的数是 121。由于没有超出计算范围，所以 Q0.0 输出为“1”。假设 IN1 中的整数为 1000，IN2 中的整数为 1000，由于乘积超出范围，所以 Q0.0 输出为“0”。




程序段1：标题：

```
        A     I     0.0
        JNB   _001
        L     MW    0
        L     MW    2
        *I
        T     MW    4
        AN    OV
        SAVE
        CLR
_001:   A     BR
        =     Q     0.0
```

图 4-111　整数乘（MUL_I）指令示例

双整数乘（MUL_DI）指令与整数乘（MUL_I）指令类似，只不过其数据类型为双整数，在此不赘述。

（4）双整数除（DIV_DI）

当允许输入端 EN 为高电平“1”时，输入端 IN1 中的双整数除以 IN2 中的双整数，结果送入 OUT 中。如果该结果超出了整数（32 位）允许的范围，OV 位和 OS 位将为“1”并且 ENO 为逻辑“0”，这样便不执行此数学框后 ENO 连接的其他函数。IN1 和 IN2 中的数可以是常数。双整数除（DIV_DI）指令和参数见表 4-50。

**表 4-50　双整数除（DIV_DI）指令和参数**

| LAD | 参　数 | 数 据 类 型 | 说　明 | 存 储 区 |
|---|---|---|---|---|
| DIV_DI<br>EN　ENO<br>IN1<br>IN2　OUT | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN1 | DINT | 被除数 | |
| | IN2 | DINT | 除数 | |
| | OUT | DINT | 除法的双整数结果（商） | |

用一个例子来说明双整数除（DIV_DI）指令，梯形图和指令表如图 4-112 所示。当 I0.0 闭合时，激活双整数除指令，IN1 中的又整数存储在 MD0 中，假设这个数为 L#11，IN2 中的双整数存储在 MD4 中，假设这个数为 L#2，双整数相除的结果存储在 OUT 端的 MD8 中的数是 L#5，不产生余数。由于没有超出计算范围，所以 Q0.0 输出为“1”。

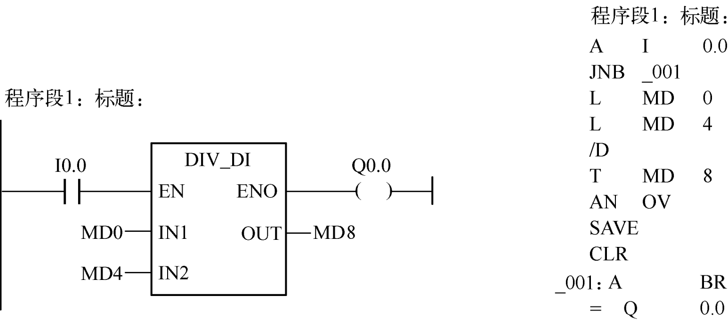



图 4-112　双整数除（DIV_DI）指令示例

**【关键点】** 双整数除法不产生余数。

整数除（DIV_I）指令与双整数除（DIV_DI）指令类似，只不过其数据类型为整数，在此不赘述。

（5）返回双整数余数（MOD_DI）

当允许输入端 EN 为高电平“1”时，输入端 IN1 中的双整数除以 IN2 中的双整数，余数送入 OUT 中。IN1 和 IN2 中的数可以是常数。返回双整数余数指令和参数见表 4-51。

**表 4-51　返回双整数余数（MOD_DI）指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| MOD_DI<br>EN　ENO<br>IN1<br>IN2　OUT | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN1 | DINT | 被除数 | |
| | IN2 | DINT | 除数 | |
| | OUT | DINT | 除法的整数结果（商） | |

用一个例子来说明返回双整数余数指令，梯形图和指令表如图 4-113 所示。当 I0.0 闭合时，激活返回双整数余数指令，IN1 中的整数存储在 MD0 中，假设这个数为 L#11，IN2 中的整数存储在 MD4 中，假设这个数为 L#2，双整数相除的余数存储在 OUT 端的 MD8 中的数是 L#1。由于没有超出计算范围，所以 Q0.0 输出为“1”。

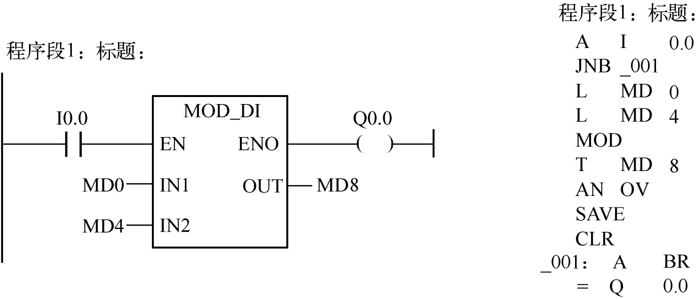



图 4-113　返回双整数余数指令示例

**2. 浮点数运算**

浮点数函数有浮点算术运算函数、三角函数函数和反三角函数、对数函数、幂函数和绝对值函数等。浮点算术函数又分为加法运算、减法运算、乘法运算和除法运算函数。浮点数运算函数见表 4-52。

**表 4-52　浮点数运算函数（部分）**

| 语　句　表 | 梯　形　图 | 描　　述 |
|---|---|---|
| +R | ADD_R | 将累加器 1，2 中的浮点数相加，浮点数运算结果在累加器 1 中 |
| -R | SUB_R | 累加器 2 中的浮点数减去累加器 1 中的浮点数，运算结果在累加器 1 中 |
| *R | MUL_R | 将累加器 1，2 中的浮点数相乘，浮点数乘积在累加器 1 中 |
| /R | DIV_R | 累加器 2 中的浮点数除以累加器 1 中的浮点数，商在累加器 1 中 |
| ABS | ABS | 取累加器 1 中的浮点数的绝对值 |

（1）实数加（ADD_R）

当允许输入端 EN 为高电平“1”时，输入端 IN1 和 IN2 中的实数相加，结果送入 OUT 中。如果该结果超出了允许的范围，OV 位和 OS 位将为“1”并且 ENO 为逻辑“0”，这样便不执行此数学框后 ENO 连接的其他函数。IN1 和 IN2 中的数可以是常数。实数加的表达式是：IN1 + IN2 = OUT。

实数加（ADD_R）指令和参数见表 4-53。

**表 4-53　实数加（ADD_R）指令和参数**

| LAD | 参　　数 | 数 据 类 型 | 说　　明 | 存　储　区 |
|---|---|---|---|---|
| ADD_R<br>EN　ENO<br>IN1<br>IN2　OUT | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN1 | REAL | 相加的第 1 个值 | |
| | IN2 | REAL | 相加的第 2 个值 | |
| | OUT | REAL | 相加的结果 | |

用一个例子来说明实数加（ADD_R）指令，梯形图和指令表如图 4-114 所示。当 I0.0 闭合时，激活实数加指令，IN1 中的实数存储在 MD0 中，假设这个数为 10.1，IN2 中的实数存储在 MD4 中，假设这个数为 21.1，实数相加的结果存储在 OUT 端的 MD8 中的数是 31.2。由于没有超出计算范围，所以 Q0.0 输出为“1”。

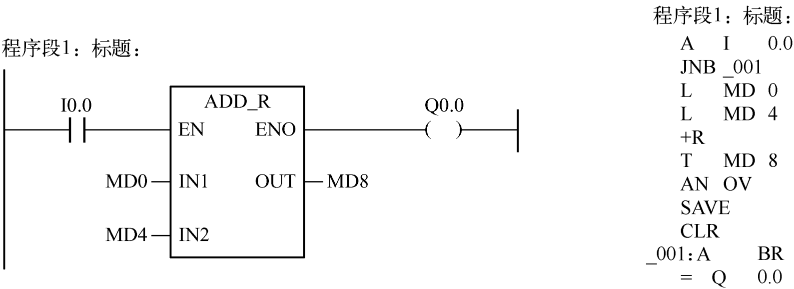



图 4-114　实数加（ADD_R）指令示例

实数减（SUB_R）指令、实数乘（MUL_R）指令和实数除（DIV_R）指令的使用方法与前面的指令用法类似，在此不赘述。

**【例 4-27】** 将 53 英寸（in）转换成以毫米（mm）为单位的整数，请设计梯形图。

**解：** 1 in = 25.4 mm，涉及到实数乘法，先要将整数转换成双整数，再将双整数转化成实数，用实数乘法指令将 in 为单位的长度变为以 mm 为单位的实数，最后四舍五入即可，梯形图如图 4-115 所示。

图 4-115 的梯形图，没有错误，但占用了较多的位存储器，在实际工作中，可以用临时变量代替时，就尽量不要使用位存储器，改进后的程序如图 4-116 所示。注意：OB1 中的 LB0 ~ LB19 被系统占用，因此本例程序使用的是从 LB30 开始的临时存储器。

（2）浮点数的绝对值（ABS）

当允许输入端 EN 为高电平“1”时，对输入端 IN 求绝对值，结果送入 OUT 中。IN 中的数可以是常数。浮点数的绝对值（ABS）的表达式是：OUT = | IN |。

浮点数的绝对值（ABS）指令和参数见表 4-54。

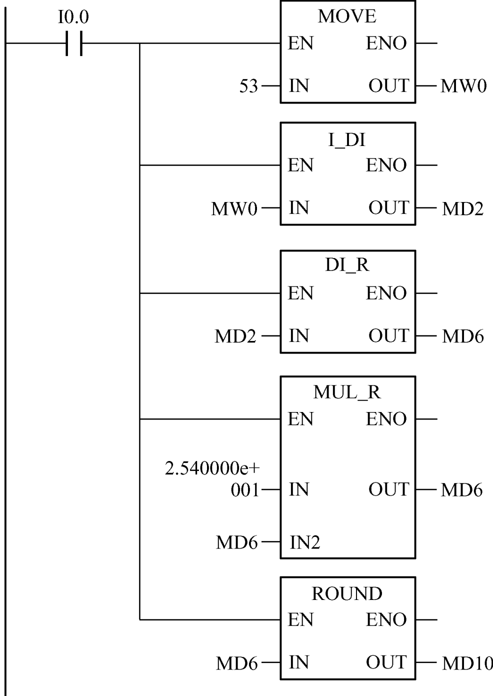



图 4-115　梯形图

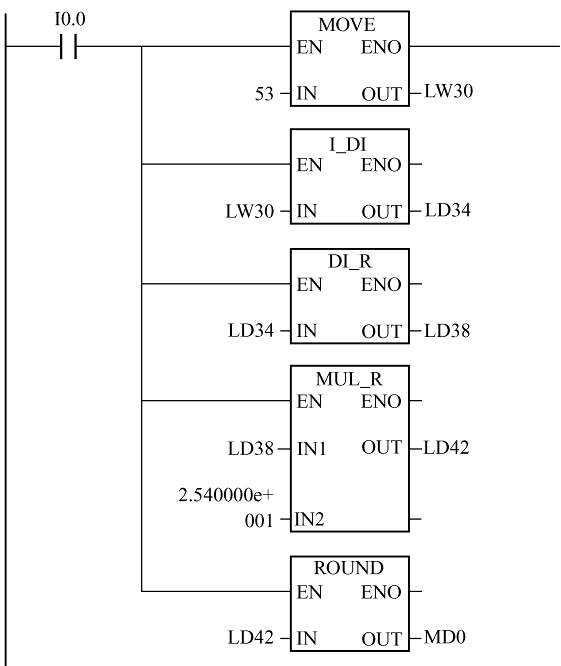
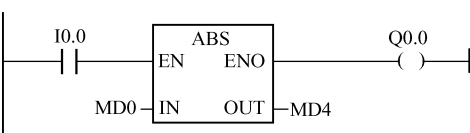



图 4-116　梯形图

**表 4-54　浮点数的绝对值（ABS）指令和参数**

| LAD | 参　数 | 数 据 类 型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| ABS<br>EN ENO<br>IN OUT | EN | BOOL | 允许输入 | I、Q、M、D、L |
| | ENO | BOOL | 允许输出 | |
| | IN | REAL | 输入值 | |
| | OUT | REAL | 输出值（绝对值） | |

用一个例子来说明浮点数的绝对值（ABS）指令，梯形图和指令表如图 4-117 所示。当 I0. 0 闭合时，激活浮点数的绝对值指令，IN 中的实数存储在 MD0 中，假设这个数为 10. 1，实数求绝对值的结果存储在 OUT 端的 MD4 中的数是 10. 1，假设 IN 中的实数为 -10. 1，实数求绝对值的结果存储在 OUT 端的 MD4 中的数是 10. 1。由于没有超出计算范围，所以 Q0. 0 输出为“1”。




程序段1：标题：

```
A     I   0.0
JNB   _001
L     MD  0
ABS
T     MD  4
SET
SAVE
CLR
_001:A      BR
=        Q   0.0
```

图 4-117　浮点数的绝对值（ABS）指令示例

### 4.5.6 逻辑控制指令

逻辑控制指令是指逻辑块中的跳转和循环指令。在没有执行跳转和循环指令之前，各语句按照先后顺序执行，也就是线型扫描。而逻辑控制指令终止了线型扫描，跳转到地址标号（Label）所指的地址，程序再次开始线型扫描。逻辑控制指令没有参数，只有一个地址标号，地址标号的作用如下：

（1）逻辑转移指令的地址是一个地址标号

地址标号最多由 4 个字母组成，第一个字符是字母，后面的字符可以是字母或者字符。

（2）目的地址标号必须从一个网络开始

跳转指令有几种形式，即无条件跳转、多分支跳转指令、与 RLO 和 BR 有关的跳转指令、与信号状态有关的跳转指令、与条件码 CC0 和 CC1 有关的跳转指令。逻辑控制指令见表 4–55。

**表 4–55 逻辑控制指令**

| 指　令 | 状态位触点指令 | 说　明 |
|---|---|---|
| JU | – | 无条件跳转 |
| JL | – | 多分支跳转 |
| JC | – | RLO =1 时跳转 |
| JCN | – | RLO =0 时跳转 |
| JCB | – | RLO =1 时跳转，再对 BR 位置位 |
| JNB | – | RLO =0 时跳转，再对 BR 位置位 |
| JBI | BR | BR =1 时跳转 |
| JNBI | – | BR =0 时跳转 |
| JO | OV | OV =1 时跳转 |
| JOS | OS | OS =1 时跳转 |
| JZ | ==0 | 运算结果为 0 时跳转 |
| JN | < > 0 | 运算结果非 0 时跳转 |
| JP | > 0 | 运算结果为正时跳转 |
| JM | < 0 | 运算结果为负时跳转 |
| JPZ | >=0 | 运算结果大于等于 0 时跳转 |
| JMZ | <=0 | 运算结果小于等于 0 时跳转 |
| JUO | UO | 指令出错时跳转 |
| LOOP | – | 循环指令 |

以下介绍几个常用的逻辑控制指令。

**1. JU，无条件跳转**

JU <跳转标签>中断线性程序扫描，并跳转到一个跳转目标，与状态字的内容无关。以下的指令表程序是 JU 指令一个应用。

```
程序段 1:标题:
        JU      LAB1                //无条件跳转到 LAB1
程序段 2:标题:
        A       I                   0.0
        =       Q                   0.0
程序段 3:标题:
LAB1:A          I                   0.2
        =       Q                   0.2
```

以上指令表对应的梯形图如图 4-118 所示。

**2. JC，当 RLO =1 时跳转**

当逻辑运算的结果为 1 时，JC <跳转标签>就中断线性程序扫描，并跳转到一个跳转目标。以下的指令表程序是 JC 指令一个应用。

```
程序段 1:标题:
        A       I       0.0         //与 I0.0
        JC      LAB1                //当 I0.0 闭合时,RL0 =1,跳转到 LAB1 程序段 2:标题:
        A       I       0.1         //当 I0.0 断开闭合时,在此继续执行程序扫描
        =       Q       0.0
程序段 3:标题:
LAB1:A          I       0.2
        =       Q       0.2
```

以上指令表对应的梯形图如图 4-119 所示。

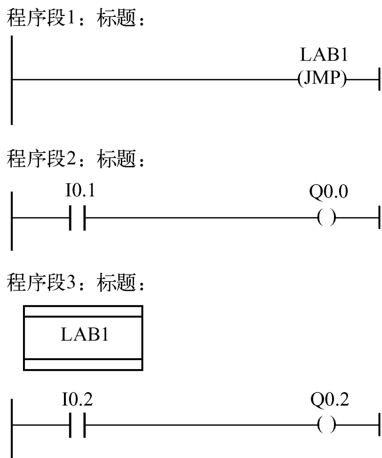



图 4-118　无条件跳转指令示例

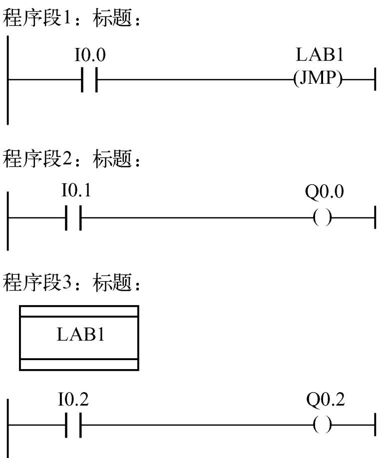



图 4-119　条件跳转指令示例

**3. JCN，当 RLO =0 时跳转**

当逻辑运算的结果为 0 时，JCN <跳转标签>就中断线性程序扫描，并跳转到一个跳转目标。以下的指令表程序是 JCN 指令一个应用。

```
            A       I   1.0
            A       I   1.2
            JCN     JOVR            //当 RLO =0 时,跳转到跳转标签 JOVR
            L       IW8             //当不执行跳转时,在此继续执行程序扫描
            T       MW22
J0VR:   A       I   2.1             //跳转到跳转标签 JOVR 后,在此继续执行程序扫描
```

**4. JCB，RLO =1 时跳转，再对 BR 位置位**

当逻辑运算的结果为 1 时，JCB <跳转标签>就中断线性程序扫描，并跳转到一个跳转

目标，再对 BR 位置位。以下的指令表程序是 JCB 指令一个应用。

```
        A     I  1.0
        A     I  1.2
        JCN   JOVR   //当 RLO =1 时,跳转到跳转标签 JOVR。将 RLO 位的内容复制到 BR 位
        L     IW8    //当不执行跳转时,在此继续执行程序扫描
        T     MW22
J0VR:  A     I  2.1  //跳转到跳转标签 JOVR 后,在此继续执行程序扫描
```

**5. JNB ，RLO =0 时跳转,，再对 BR 位置位**

当逻辑运算的结果为 0 时，JNB <跳转标签>就中断线性程序扫描，并跳转到一个跳转目标，再对 BR 位置位。以下的指令表程序是 JNB 指令一个应用。

```
        A     I  1.0
        A     I  1.2
        JCN   JOVR   //当 RLO =0 时,跳转到跳转标签 JOVR。将 RLO 位的内容复制到 BR 位中
        L     IW8    //当不执行跳转时,在此继续执行程序扫描
        T     MW22
J0VR:  A     I  2.1  //跳转到跳转标签 JOVR 后,在此继续执行程序扫描
```

**6. LOOP 循环**

LOOP <跳转标签>（对 ACCU1 的低位 进行减 1 操作，并在 ACCU1 < >0 时跳转）可简化循环编程。ACCU1 中包含循环计数器。指令跳转到指定的跳转目标。只要 ACCU1 的内容不等于 0，就一直执行跳转。线性程序扫描在跳转目标处继续执行。以下的指令表程序是 LOOP 指令一个应用。

```
        L       1#1       //将整型常数(32 位)装载到 ACCU1 中
        T       MD2.0     //将 ACCU1 的内容传送给 MD20(初始化)
        L       5         //将循环周期的数目装载到 ACCU1 – L 中
NEXT:   T       MW1.0     //跳转标签 = 循环开始/将 ACCU1 – L 传送给循环计数器
        L       MD2.0
        *       D         //MD2.0 的当前内容乘以 MB1.0 的当前内容
        T       MD2.0     //将相每间结果传送给 MD2.0
        L       MW1.0     //将循环计数器的内容装载到 ACCU1 中
        LOOP    NEXT      //对 ACCU1 的内容进行减 1 操作,当 ACCU1 – L >0 时
                          //跳转到 NEXT 跳转标签
        L       MW24      //完成循环后,在此继续执行程序扫描
        L       200
        >I
```

## 4.5.7 字逻辑运算指令

字的逻辑运算指令一共有 6 条，分别是：字与运算（WAND_W）、字或运算（WOR_W）、字异或运算（WXOR_W）、双字与运算（WAND_DW）、双字或运算（WOR_DW）、双字异或运算（WXOR_DW）。

字逻辑指令就是对 16 字或者 32 双字逐位进行逻辑运算，一个操作数在累加器 1，另一个操作数在累加器 2，指令中也允许有立即数（常数）的形式输出。以下用一个例子介绍字与运算（WAND_W）的应用。需要把 MW0 把 QW0，但 QW0 的低 4 位要清零。梯形图程序如图 4–120 所示。

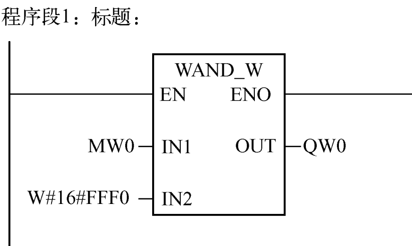



图 4–120　梯形图

以上梯形图也可指令表表达，如下所示：

```
L    MW          0        //将 MW0 转入累加器 1
L    W#16#FFF    0        //将累加器的内容装入累加器 2,将 W#16#FFFF0 装入累加器 1
AW                        //将累加器 1 和累加器 2 逐位与运算
T    QW          0        //将与运算的结果存入 QW0
```

## 4.5.8 其他指令

### 1. 主控继电器指令

主控继电器（Master Control Relay），简称 MCR，主控继电器指令用来控制 MCR 区内的指令是否被正常执行，相当于一个用于接通和断开“能流”主令开关。主控继电器指令并不常用。

MCRA 为激活主令继电器指令，MCRD 为结束主令继电器指令，MCR < 是主控制继电器打开，MCR > 是主控制继电器关闭，以下用一个例子介绍主控继电器指令。

如图 4-121 所示的梯形图，当 I0.0 闭合时，才执行“MCR <”到“MCR >”指令之间的程序，否则跳过该段程序。

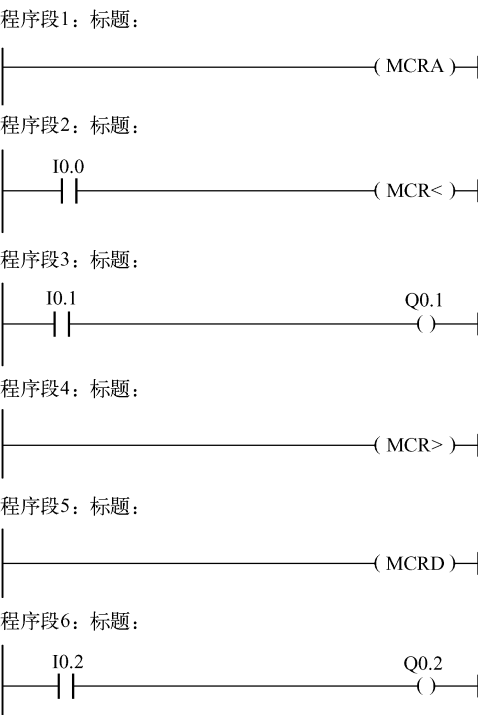



图 4-121 梯形图

此外，主控继电器指令最多可以 8 级嵌套。

### 2. 数据块指令

数据块指令见表 4-56，在访问数据块时，先要指明要访问哪一个数据块，以及访问数据块的哪一个存储单元的地址。指令如果同时给出数据块的编号和数据在数据块中的地址（例如 DB1. DBW0），就可以使用绝对地址访问，也可以使用符号地址访问。

表 4-56　数据块指令

| 序　　号 | 指　　令 | 功 能 说 明 |
| --- | --- | --- |
| 1 | OPN | 打开数据块 |
| 2 | CDB | 交换共享数据块和背景数据块的编号 |
| 3 | L DBLG | 共享数据块的长度装入累加器 1 |
| 4 | L DBNO | 共享数据块的编号装入累加器 1 |
| 5 | L DILG | 背景数据块的长度装入累加器 1 |
| 6 | L DINO | 背景数据块的编号装入累加器 1 |

数据块指令应用的指令表程序如下所示。

```
OPN   DI    1          //打开背景数据块 DB1
L     DIW   0          //将 DB1. DIWO 装入累加器 1
OPN   DB    2          //打开共享数据块 DB2
T     DBW   2          //将累加器 1 装入 DB2. DBW2
```

数据块在工程中极为常用，而且有一定难度，在后续章节还要进行详细讲解。

**3. 交换指令（TAK）**

就是把累加器 1 和累加器 2 的内容互换。

**【例 4-28】** 设计一段程序，把 MW0 和 MW2 中较大的一个减去较小的一个，结果存入 MW8 中。

**解**：指令表程序如下：

```
       L     MW    0     //将 MW0 装入累加器 1
       L     MW    2     //将累加器 1 的内容装入累加器 2,再将 MW2 装入累加器
       >I                //如 MWO 大于 MW2,则跳转到 LAB1
       JC    LAB1
       TAK               //累加器 1 和累加器 2 的内容互换
LAB1：-I                 //累加器 2 减去累加器 1 的内容
       T     MW    8     //将运算结果存入 MW8
```

**4. 8 位加（INC）、8 位减指令（DEC）**

字节加指令 INC 就是将累加器 1 的低字节加一个 8 位的常数（其大小范围是 0～255），结果仍然保存在累加器 1 的低字节；字节减指令 DEC 就是将累加器 1 的低字节减一个 8 位的常数（其大小范围是 0～255），结果仍然保存在累加器 1 的低字节。以下用一个简单的指令说明 DEC 的用法。

```
L     MB    0          //将 MB0 装入累加器 1
DEC   2                //将累加器 1 的内容减去 2
T     MB    1          //将运算结果存入 MB1
```

**5. 空操作指令（BLD、NOP）**

BLD <number> 是程序显示指令，只用于编程设备的图形显示，不执行任何功能。

NOP 0 是空操作指令，不执行任何功能，也不会影响状态位。

## 4.6　实例

至此，读者已经对 S7－300/400 PLC 的软硬件有了一定的了解，本节内容将列举一些简

单的例子，供读者模仿学习。

## 4.6.1　电动机的控制

**【例 4-29】**请设计电动机点动控制的梯形图和接线图。

**解：**

（1）方法 1

常规设计方法的原理图和梯形图如图 4-122 和 4-123 所示。但如果程序用到置位指令（S Q0.0），则这种解法则不可用。

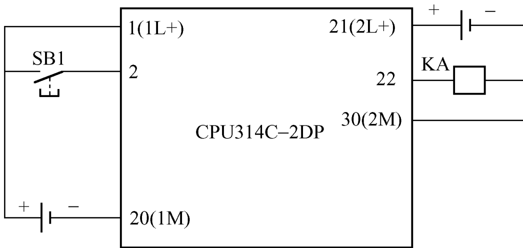



图 4-122　原理图

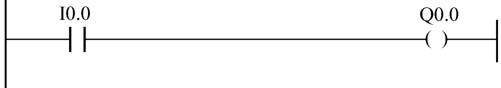



图 4-123　梯形图

（2）方法 2，如图 4-124。

**【例 4-30】**请设计两地控制电动机的启停的梯形图和接线图。

**解：**

（1）方法 1

常规设计方法的原理图和梯形图如图 4-125 和 4-126 所示。这种解法是正确的解法，但不是最优方案，因为这种解法占用了较多的 I/O 点。

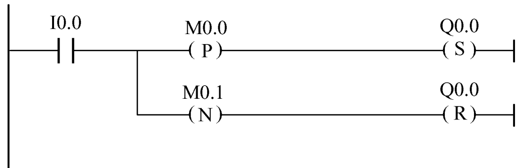



图 4-124　梯形图

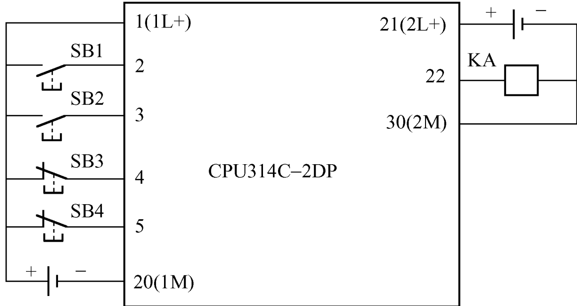



图 4-125　原理图

（2）方法 2，如图 4-127。

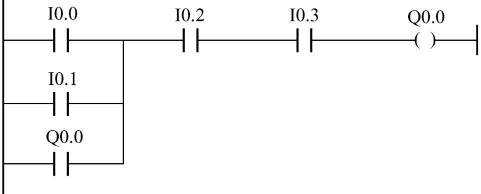



图 4-126　梯形图

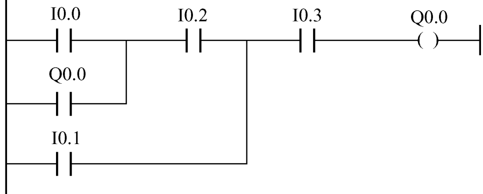



图 4-127　梯形图

（3）方法 3

优化后的方案的原图如图 4-128 所示，梯形图如图 4-129 所示。可见节省了两个输入点，但功能完全相同。

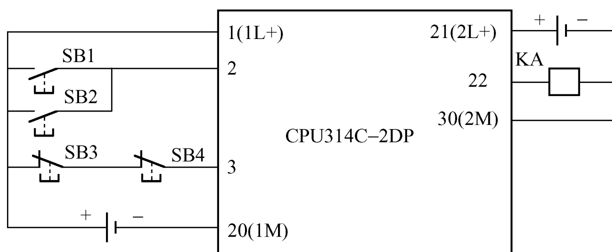



图 4-128　原理图

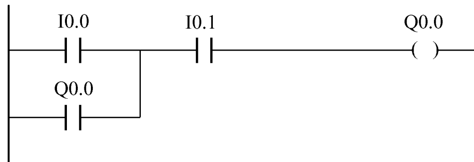



图 4-129　梯形图

**【例 4-31】**请编写电动机的起动优先的控制程序。

**解：**I0. 0 是起动按钮接常开触点，I0. 1 是停止按钮接常闭触点。起动优先于停止的程序如图 4-130 所示。优化后的程序如图 4-131 所示。

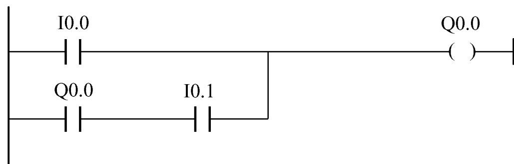



图 4-130　梯形图

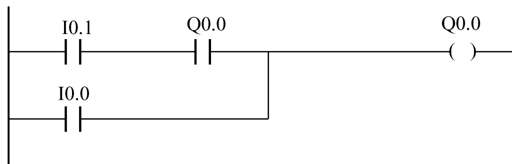



图 4-131　梯形图

**【例 4-32】**请编写程序，实现电动机的启/停控制和点动控制，请画出梯形图和接线图。

**解：**输入点：起动 - I0. 0，停止 - I0. 2，点动 - I0. 1；

输出点：正转 - Q0. 0。

接线图如图 4-132，梯形图如图 4-133 所示，这种编程方法在工程实践中较为常用。

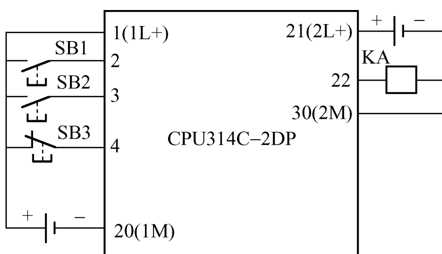



图 4-132　原理图

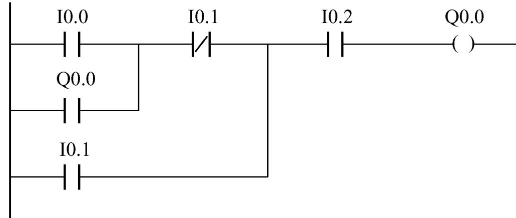



图 4-133　梯形图

以上程序还可以用如图 4-134 所示的梯形图程序替代。

**【例 4-33】**设计电动机的“正转 - 停 - 反转”的梯形图，其中 I0. 0 是正转按钮、I0. 1 是反转按钮、I0. 2 是停止按钮、Q0. 0 是正转输出、Q0. 1 是反转输出。

**解：**先设计 PLC 的 I/O 接线图，如图 4-135 所示。

借鉴继电器接触器系统中的设计方法，不难设计“正转－停－反转”梯形图，如图4-136所示。常开触点Q0.0和常开触点Q0.1起自保（自锁）作用，而常闭触点Q0.0和常闭触点Q0.1起互锁作用。

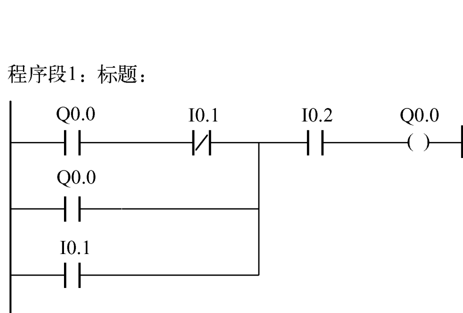



图4-134　梯形图

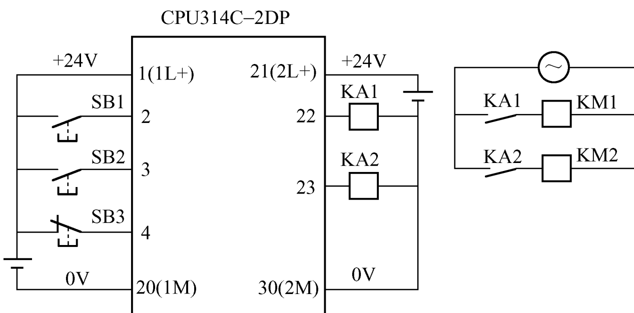



图4-135　I/O接线图

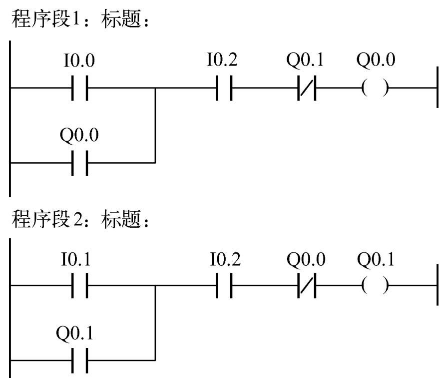



图4-136　“正转－停－反转”梯形图

## 4.6.2　定时器和计数器应用

**【例4-34】** 请编写一段程序，实现分脉冲功能。

**解：** 解题思路：先用定时器产生秒脉冲，再用30个秒脉冲作为高电平，30个脉冲作为低电平，秒脉冲用“周期/时钟存储器”的M0.5产生，其硬件组态如图4-137所示。梯形图如图4-138所示。

此题的另一种解法如图4-139所示，请读者思考，图4-139的梯形图的程序段1和程序段2的顺序后，是否可行？为什么？

**【例4-35】** 某设备，三条传送带上的工件，最后汇集到最后一条传送到上，前三条传送带上有I0.0、I0.1和I0.2三个接近开关，先到的工件感应接近开关，激发对应的气缸动作，推动工件，前三条传送带对应的气缸输出为Q0.0、Q0.1和Q0.2，每次只能保证一条生产的一个工件送到最后一条传送带上（竞争上岗），工件送到最后一条传送带，1s后才能送下一个工件，复位为I0.3。请编写相关程序实现此功能。

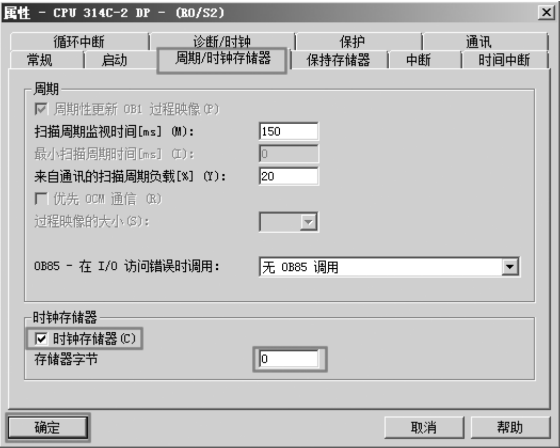



图 4-137　硬件组态

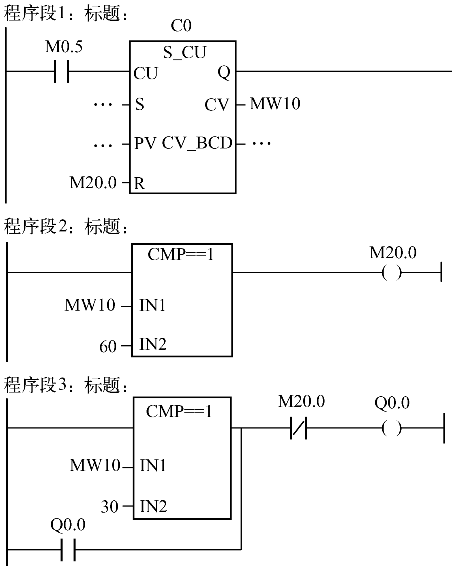



图 4-138　梯形图

**解：**梯形图程序如图 4-140 所示。

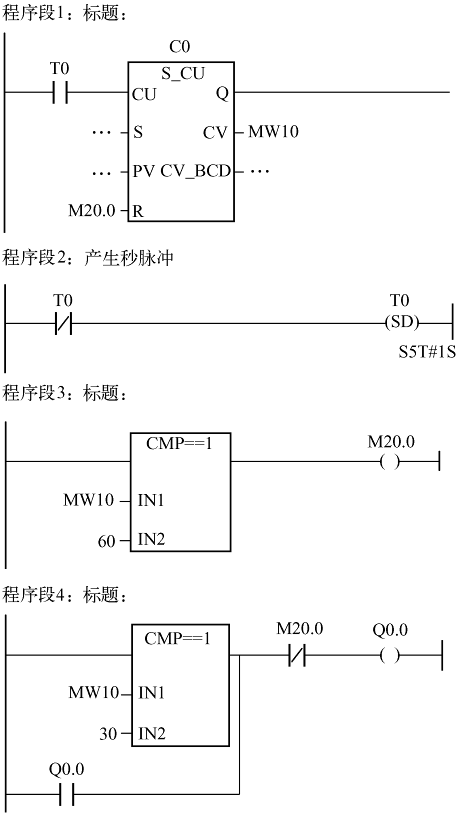



图 4-139　梯形图

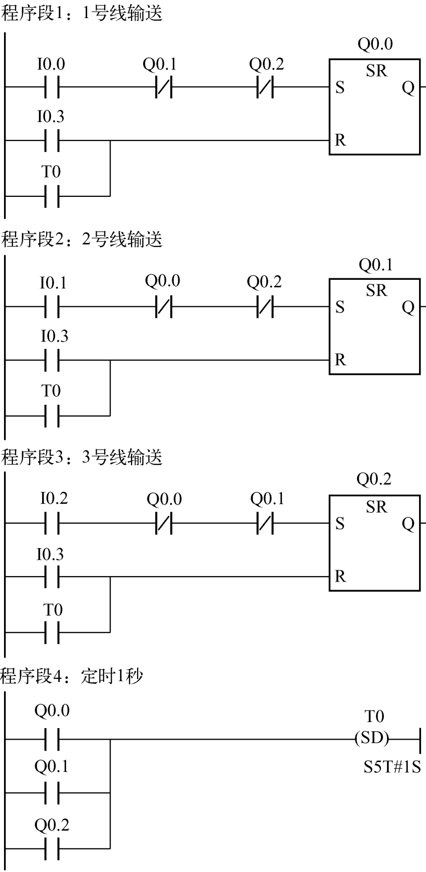



图 4-140　梯形图

【例 4-36】请设计一段程序，延时时间为 100 h。

解：梯形图程序如图 4-141 所示。

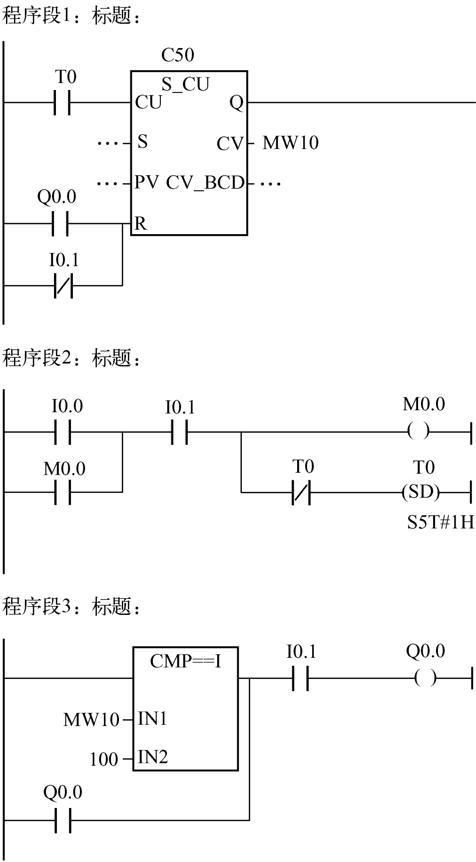



图 4-141　梯形图

请读者思考，图 4-141 梯形图的程序段 1 和程序段 2 顺序调换，是否可行？为什么？

# 第 5 章　S7－300/400 PLC 的程序结构

本章主要介绍功能、功能块、系统功能、系统功能块、数据块、中断和组织块等，学习本章内容，对编写程序至关重要。

## 5.1　STEP 7 编程方法简介

STEP 7 编程方法有三种：线性化编程、模块化编程和结构化编程。以下对这三种方法分别进行简介。

**1. 线性化编程**

线性化编程就是将整个程序放在循环控制组织块 OB1 中，CPU 循环扫描执行 OB1 中的全部指令。其特点是结构简单、概念简单，但由于所有指令都在一个块中，程序的某些部分可能不需要多次执行，而扫描时，重复扫描所有的指令，会造成资源浪费、执行效率低。对于大型的程序要避免线性化编程。

**2. 模块化编程**

模块化编程就是将程序根据功能分为不同的逻辑块，每个逻辑块完成不同的功能。在 OB1 中可以根据条件调用不同的功能或者功能块。其特点是易于分工合作，调试方便。由于逻辑块是有条件调用，所以提高了 CPU 的效率。

**3. 结构化编程**

结构化编程就是将过程要求中类似或者相关的任务归类，在功能或者功能块中编程，形成通用的解决方案。通过不同的参数调用相同的功能或者通过不同的背景数据块调用相同的功能块。一般而言，S7－300/400 PLC 的程序都不是小型程序，所以通常采用结构化编程方法。

结构化编程具有如下一些优点：

- 各单个任务块的创建和测试可以相互独立地进行。
- 通过使用参数，可将块设计得十分灵活。比如，可以创建一钻孔循环，其坐标和钻孔深度可以通过参数传递进来。
- 块可以根据需要在不同的地方以不同的参数数据记录进行调用，也就是说，这些块能够被再利用。
- 在预先设计的库中，能够提供用于特殊任务的“可重用”块。

## 5.2　功能、数据块和功能块

### 5.2.1　块的概述

**1. 块的简介**

在操作系统中包含了用户程序和系统程序，操作系统已经固化在 CPU 中，它提供 CPU

运行和调试的机制。CPU 的操作系统是按照事件驱动扫描用户程序的。用户程序写在不同的块中，CPU 按照执行的条件成立与否执行相应的程序块或者访问对应的数据块。用户程序则是为了完成特定的控制任务，是由用户编写的程序。用户程序通常包括组织块（OB）、功能块（FB）、功能（FC）和数据块（DB）。系统块包括系统功能（SFC）、系统功能块（SFB）和系统数据块（SDB）。用户程序中的块的说明见表 5-1。

**表 5-1　用户程序中的块的说明**

| 块的类型 | 属　　性 |
|---|---|
| 组织块（OB） | • 用户程序接口<br>• 优先级（0～27）<br>• 在局部数据堆栈中指定开始信息 |
| 功能（FC） | • 参数可分配（必须在调用时分配参数）<br>• 基本上没有存储空间（只有临时变量） |
| 功能块（FB） | • 参数可分配（可以在调用时分配参数）<br>• 具有（收回）存储空间（静态变量） |
| 数据块（DB） | • 结构化的局部数据存储（背景数据块 DB）<br>• 结构化的全局数据存储（在整个程序中有效） |
| 系统功能块（SFB） | • SFB 具有存储空间，存储在 CPU 的操作系统中并可由用户调用 |
| 系统功能（SFC） | • SFC 无存储空间，存储在 CPU 的操作系统中并可由用户调用 |
| 系统数据块（SDB） | • 用于配置数据和参数的数据块 |

**2. 块的结构**

块由变量声明表和程序组成。每个逻辑块都有变量声明表，变量声明表是用来说明块的局部数据。而局部数据包括参数和局部变量两大类。在不同的块中可以重复声明和使用同一局部变量，因为它们在每个块中仅一次有效。

局部变量包括两种：静态变量和临时变量。

参数是在调用块与被调用块之间传递的数据，包括输入、输出和输入/输出变量。表 5-2 为局部数据声明类型。

**表 5-2　局部数据声明类型**

| 变量名称 | 变量类型 | 说　　明 |
|---|---|---|
| 输入 | IN | 为调用模块提供数据，输入给逻辑模块 |
| 输出 | OUT | 从逻辑模块输出数据结果 |
| 输入/输出 | IN_OUT | 参数值既可以输入，也可以输出 |
| 静态变量 | STAT | 静态变量存储在背景数据块中，块调用结束后，变量被保留 |
| 临时变量 | TEMP | 临时变量存储 L 堆栈中，块执行结束后，变量消失 |

图 5-1 所示为块调用的分层结构的一个例子，组织块 OB1（主程序）调用功能块 FB1，FB1 调用功能块 FB10，组织块 OB1（主程序）调用功能块 FB2，功能块 FB2 调用功能 FC5，功能 FC5 调用系统功能 SFC0。

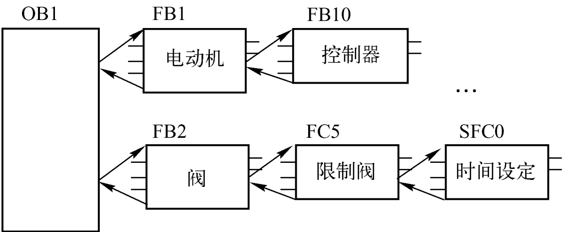



图 5-1　块调用的分层结构

## 5.2.2　功能（FC）及其应用

**1. 功能（FC）简介**

1）功能（FC）是用户编写的程序块。功能是一种“不带内存”的逻辑块。属于 FC 的临时变量保存在本地数据堆栈中。执行 FC 时，该数据将丢失。为永久保存该数据，FC 也可使用共享数据块。由于 FC 本身没有内存，因此，必须始终给它指定实际参数。不能给 FC 的本地数据分配初始值。

2）FC 里有一个局域变量表和块参数。局域变量表里有：IN（输入参数）、OUT（输出参数）、IN_OUT（输入/输出参数）、TEMP（临时数据）、RETURN（返回值 RET_VAL）。IN（输入参数）将数据传递到被调用的块中进行处理。OUT（输出参数）是将结果传递到调用的块中。IN_OUT（输入/输出参数）将数据传递到被调用的块中，在被调用的块中处理数据后，再将被调用的块中发送的结果存储在相同的变量中。TEMP（临时数据）是块的本地数据，并且在处理块时将其存储在本地数据堆栈。关闭并完成处理后，临时数据就变得不再可访问。RETURN 包含返回值 RET_VAL。

**2. 功能（FC）的应用**

功能（FC）类似于 VB 语言中的子程序，用户可以将具有相同控制过程的程序编写在 FC 中，然后在主程序 OB1 中调用。创建功能的过程步骤是：先建立一个项目，再在 SIMATIC 管理器界面中选中“块”，单击菜单栏的“插入”→“S7 块”→“功能”，即可插入一个空的功能。以下用三个例题讲解功能（FC）的应用。

**【例 5-1】**用功能实现电动机的起停控制。

**解：**

1）新建一个项目，本例为“起停控制”。选中“块”，接着单击菜单栏的“插入”→“S7 块”→“功能”，即可插入一个空的功能，如图 5-2 所示。

2）如图 5-3 所示，在“属性－功能”界面中，输入功能的名称，再单击“确定”按钮。再双击“FC1”，打开功能，如图 5-4 所示。




图 5-2　插入功能




图 5-3　“属性－功能”界面

3）在“程序编辑器”中输入如图 5-5 所示的程序，此程序能实现起停控制，再保存程序。



图 5-4　打开功能

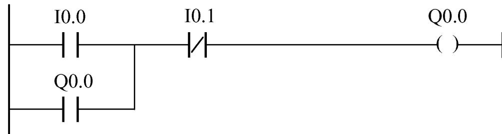



图 5-5　功能中的程序

4）在 SIMATIC 管理器界面，双击“OB1”，打开主程序块“OB1”，如图 5-6 所示。

5）将功能“FC1”拖入程序段 1，如图 5-7 所示。如果将整个项目下载到 PLC 中，即可实现“起停控制”。



图 5-6　打开主程序块



图 5-7　在主程序中调用功能

在例 5-1 中，只能用 I0.0 实现起动，而用 I0.1 实现停止，这种功能调用方式是绝对调用，显然灵活性不够，例 5-2 将用参数调用。

**【例 5-2】**用功能实现电动机的起停控制。

**解：**本例的 1）、2）步与例 5-1 相同，在此不再重复讲解。

3）在 SIMATIC 管理器中，双击功能块“FC1”，打开功能，弹出“程序编辑器”界面，先选中 IN（输入参数）新建参数“Start”和“Sto”，数据类型为“BOOL”，如图 5-8 所示。再选中 OUT（输出参数），新建参数“Motor”，数据类型为“BOOL”，如图 5-9 所示。最后在程序段 1 中输入程序，如图 5-9 所示，注意参数前都要加“#”。



图 5-8　新建输入参数

4）在 SIMATIC 管理器界面，双击“OB1”，打开主程序块“OB1”，将功能“FC1”拖入程序段 1，如图 5-10 所示。如果将整个项目下载到 PLC 中，就可以实现“起停控制”。这个程序的功能“FC1”的调用比较灵活，与例 5-1 不同，起动不只限于 I0.0，停止不只限于 I0.1，在编写程序时，可以灵活分配应用。




图 5-9　功能 FC1




图 5-10　在 OB1 中调用功能 FC1

**【例 5-3】** 某系统采集一路模拟量（温度），温度的范围是 0～200℃，要求对温度值进行数字滤波，算法是：将最近的三次采样数值相加，取平均值，即是最终温度值。

**解：**

1）数字滤波的程序是功能 FC1 先创建一个空的功能，打开功能，并创建输入参数“GatherV”，如图 5-11 所示，就是采样输入值；创建输出参数“Result1”，如图 5-12 所示，就是数字滤波的结果；创建输入/输出参数“LastV”（前一个数值）、“LastestV”（最前一个数值）和“EarlyV”（当前数值），如图 5-13 所示，输入/输出参数既可以在方框的输入端，也可以在方框的输出端，应用比较灵活；创建临时变量参数“temp1”和“temp2”，如图 5-14 所示，临时变量参数既可以在方框的输入端，也可以在方框的输出端，应用也比较灵活。




图 5-11　输入参数

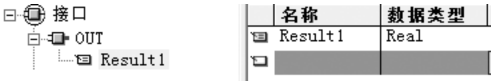



图 5-12　输出参数

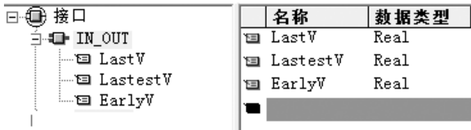


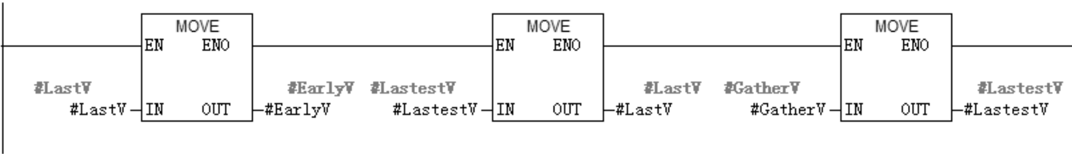

图 5-13　输入输出参数




图 5-14　临时参数

2）在 FC1 中，编写滤波梯形图程序如图 5-15 所示。

⊟ **程序段 1**：将3次采集的温度数值按照次序保存




⊟ **程序段 2**：求3次平均值

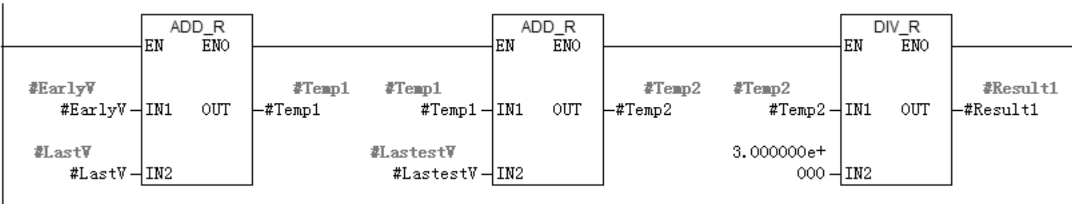
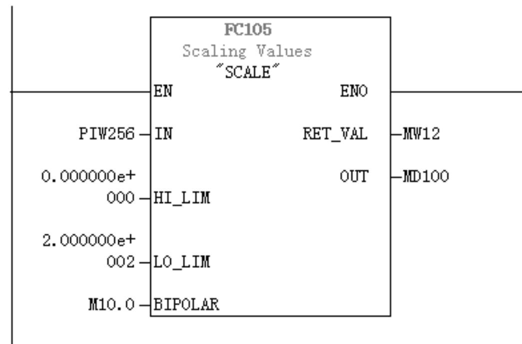



图 5-15　FC1 中的梯形图

3）在 OB1 中，编写梯形图程序如图 5-16 所示。

⊟ **程序段 1**：标题：




⊟ **程序段 2**：标题：

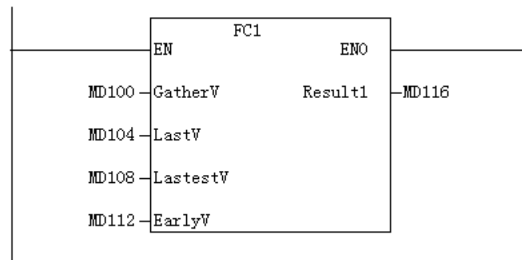



图 5-16　OB1 中的梯形图

## 5.2.3　共享数据块（DB）及其应用

### 1. 共享数据块（DB）简介

共享数据块（DB）与逻辑块不同，数据块不包含 STEP 7 指令。它们用来存储用户数据，即数据块包含用户程序使用的变量数据。共享数据块则用来存储可由所有其他块访问的

用户数据。共享数据块（DB）的应用非常广泛。

**2. 共享数据块（DB）应用**

以下用两个例题来说明数据块的应用。

【**例 5-4**】用数据块实现电动机的启停控制。

**解：**

1）新建一个项目，本例为“数据块应用”，选中“块”，单击菜单栏的“插入”→“S7 块”→“数据块”，即可插入一个空的数据块，如图 5-17 所示。

2）如图 5-18 所示，在“属性 - 数据块”界面中，输入数据块的名称，再单击“确定”按钮即可。




图 5-17　插入数据块




图 5-18　“属性 - 数据块”界面

3）在 SIMATIC 管理器界面中，选中“块”，单击菜单栏的“插入”→“S7 块”→“变量表”，即可插入一个空的变量表，如图 5-19 所示。

4）在“程序编辑器”中输入如图 5-20 所示的程序，此程序能实现起停控制，保存程序。

5）在 SIMATIC 管理器界面中，双击变量表“VAT_1”，打开变量表，并输入“1”处的地址、显示格式和修改数值，如图 5-21 所示。再将整个项目下载到 CPU 中，当单击“监视参数”按钮和“修改变量”按钮时，Q0.0 闭合，可以控制电动机运行，当把“true”修改成“false”时，电动机停止运行。




图 5-19　插入变量表




图 5-20　数据块中的程序




图 5-21　监控参数

【关键点】数据块的使用比较灵活，除了上述的 BOOL 数据类型，还有其他数据类型，如 DB1. DBB0 表示字节，DB1. DBW0 表示字，DB1. DBD0 表示双字，在后面会用到。

数组在 S7 -300/400 PLC 中较为常用，以下的例子是用数据块创建数组。

**【例 5-5】** 用数据块创建一个数组 ary[0..5]，数组中包含 6 个整数，并编写程序把模拟量 PIW752 保存到数组的第 3 个整数中。

**解：**

1）先进行硬件组态，并创建共享数组块 DB1，如图 5-22 所示，双击“DB1”打开数据块“DB1”。

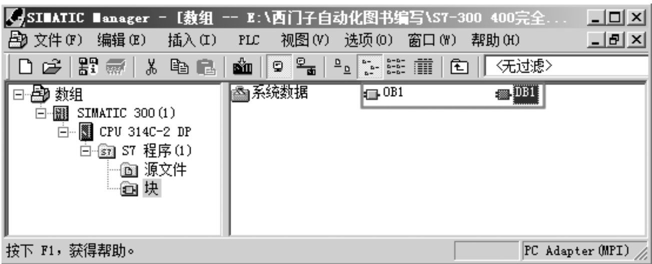



图 5-22　创建新项目

2）在 DB1 中创建数组。数组名称 ary，数组 ARRAY[0..5]表示数组中有 6 个元素，INT 表示数组的数据为整数，如图 5-23 所示，保存创建的数组。

3）在 OB1 中编写梯形图程序，如图 5-24 所示。

| 地址 | 名称 | 类型 | 初始值 | 注释 |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | ary | ARRAY[0..5] | | 数组 |
| *2.0 | | INT | | |
| =12.0 | | END_STRUCT | | |

图 5-23　创建数组

**程序段 1：标题：**

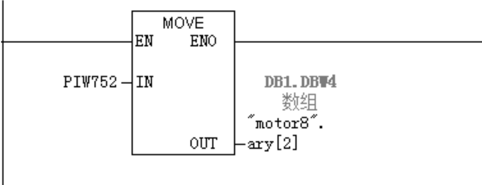



图 5-24　OB1 中梯形图

## 5.2.4　自定义数据类型（UDT）及其应用

自定义数据类型是个难点，对于初学者更加如此。虽然在前面章节已经提到了自定义数据类型，但由于前述章节的部分知识点所限，前面章节没有讲解应用。以下用一个例子介绍自定义数据类型的应用，以便帮助读者进一步理解自定义数据类型。

**【例 5-6】** 有 10 台电动机，要对其进行起停控制，而且还要采集其温度信号，请设计此控制系统，并编写控制程序（要求使用自定义数据类型）。

**解：** 每台电动机都有起动、停止、电动机和温度四个参数，因此需要创建 40 个参数，这是一种方案；但更简单的方案是先创建起动、停止、电动机和温度四个参数，再把这四个参数作为一个自定义的数据类型，每台电动机都可以使用创建的“自定义”的数据类型，而不必新建 40 个参数，这样方案更加简单。自定义数据类型在工程中是较为常用的。

1）首先新建一个项目，命名为“UDT”，并插入数据块“DB1”和自定义数据“UDT1”，如图 5-25 所示。

2）打开 UDT 的属性，将其符号命名为“Motor”，单击“确定”按钮，如图 5-26 所示。再打开 UDT1，在其中创建如图 5-27 所示的数据结构，“Motor”就是一个自定的数据类型，共有 4 个参数，这个新自定义的数据类型，可以在程序中使用。

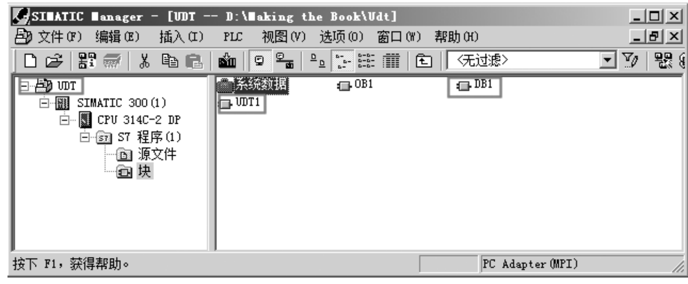



图 5-25　新建项目“UDT”，插入“DB1”和“UDT1”

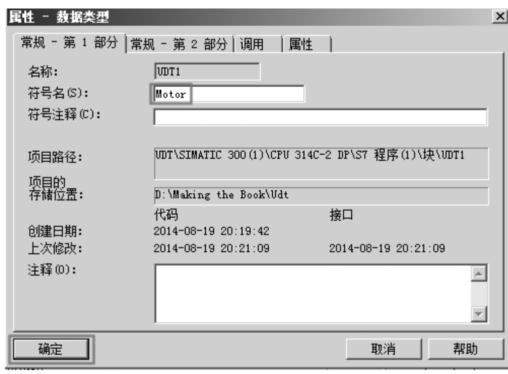



图 5-26　将 UDT 符号命名为“Motor”

| 地址 | 名称 | 类型 | 初始值 | 注释 |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | Starting | BOOL | FALSE | 启动 |
| +0.1 | Stoping | BOOL | FALSE | 停止 |
| +0.2 | Mout | BOOL | FALSE | 电动机 |
| +2.0 | Temperature | INT | 20 | 温度 |
| =4.0 | | END_STRUCT | | |

图 5-27　设置 UDT1 中的参数

3）打开 DB1 的属性，将其符号命名为“Motora”，单击“确定”按钮，如图 5-28 所示。再打开 DB1，如图 5-29 所示，创建参数“Motor1”，其数据类型为 UDT 的数据类型“Motor”。

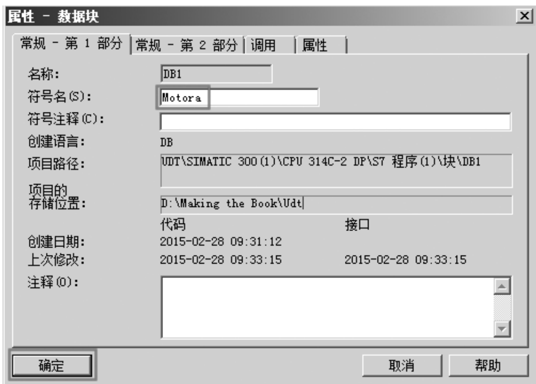



图 5-28　将 DB1 符号命名为“Motora”

| 地址 | 名称 | 类型 |
|---|---|---|
| 0.0 | | STRUCT |
| +0.0 | Motor1 | "Motor" |
| +4.0 | Motor2 | "Motor" |
| +8.0 | Motor3 | "Motor" |
| +12.0 | Motor4 | "Motor" |
| +16.0 | Motor5 | "Motor" |
| +20.0 | Motor6 | "Motor" |
| +24.0 | Motor7 | "Motor" |
| +28.0 | Motor8 | "Motor" |
| +32.0 | Motor9 | "Motor" |
| +36.0 | Motor10 | "Motor" |
| =40.0 | | END_STRUCT |

图 5-29　设置 DB1 中的参数（声明视图）

单击菜单栏中的“视图”→“数据视图”，图 5-29 变成如图 5-30 所示的详细视图。

4）编写如图 5-31 所示的梯形图程序。梯形图中用到了自定义数据类型。

| 地址 | 名称 | 类型 | 初始值 | 实际值 |
|---|---|---|---|---|
| 0.0 | Motor1.Starting | BOOL | FALSE | FALSE |
| 0.1 | Motor1.Stoping | BOOL | FALSE | FALSE |
| 0.2 | Motor1.Mout | BOOL | FALSE | FALSE |
| 2.0 | Motor1.Temperature | INT | 0 | 0 |
| 4.0 | Motor2.Starting | BOOL | FALSE | FALSE |
| 4.1 | Motor2.Stoping | BOOL | FALSE | FALSE |
| 4.2 | Motor2.Mout | BOOL | FALSE | FALSE |
| 6.0 | Motor2.Temperature | INT | 0 | 0 |
| 8.0 | Motor3.Starting | BOOL | FALSE | FALSE |
| 8.1 | Motor3.Stoping | BOOL | FALSE | FALSE |
| 8.2 | Motor3.Mout | BOOL | FALSE | FALSE |
| 10.0 | Motor3.Temperature | INT | 0 | 0 |

图 5-30　设置 DB1 中的参数（数据视图，部分）

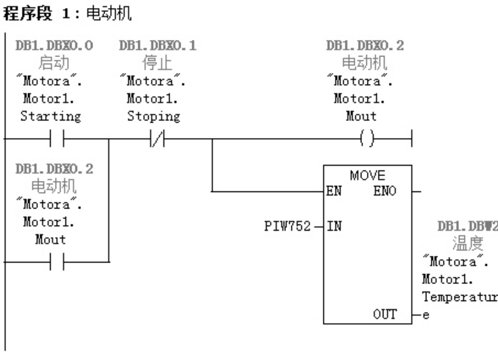



图 5-31　梯形图

## 5.2.5　功能块（FB）及其应用

### 1. 功能块（FB）的简介

功能块（FB）属于编程者自己编程的块。功能块是一种“带内存”的块。分配数据块作为其内存（数据块）。传送到 FB 的参数和静态变量保存在实例 DB 中。临时变量则保存在本地数据堆栈中。执行完 FB 时，不会丢失 DB 中保存的数据。但执行完 FB 时，会丢失保存在本地数据堆栈中的数据。

### 2. 功能块（FB）的应用

以下用一个例题来说明功能块的应用。

**【例 5-7】** 用功能块实现对一台电动机的星 – 三角起动控制。

**解：** 星 – 三角起动的电气原理图如图 5-32 所示。注意停止按钮接常闭触点。

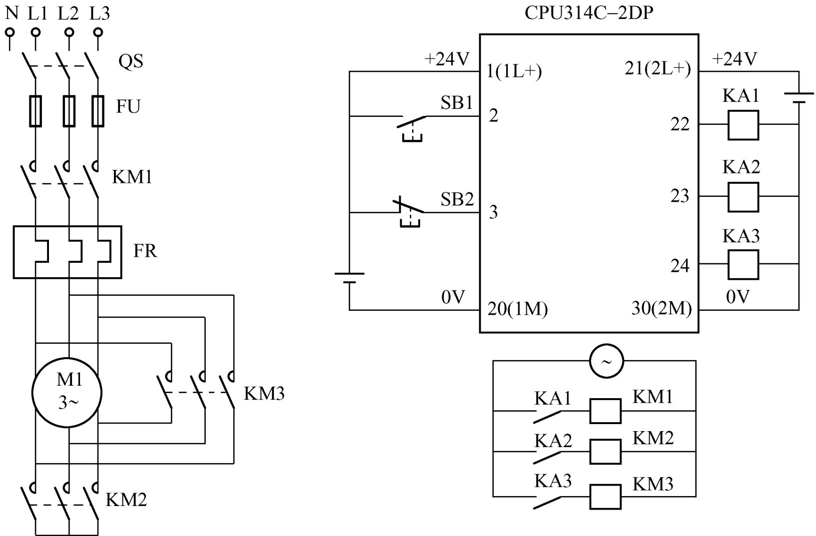



图 5-32　电气原理图

星－三角起动的项目创建如下：

1）新建一个项目，本例为“起停控制”，选中“块”，单击菜单栏的“插入”→“S7块”→“功能块”，即可插入一个空的功能块，如图5-33所示。

2）如图5-34所示，在“属性－功能块”界面中，输入功能块的名称，单击“确定”按钮。再双击“FB1”，打开功能块，如图5-35所示。




图5-33　插入功能块

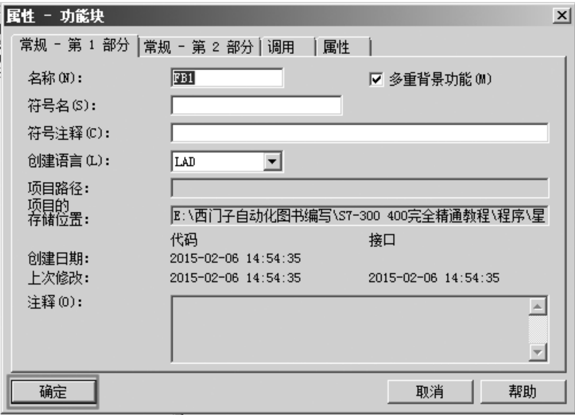



图5-34　“属性－功能块”界面

3）在接口“IN”中，新建4个变量，如图5-36所示，注意变量的类型。注释内容可以空缺，注释的内容支持汉字字符。




图5-35　打开功能块




图5-36　在接口“IN”中，新建4个变量

4）在接口“OUT”中，新建3个变量，如图5-37所示。

5）在接口“STAT”中，新建2个静态变量，如图5-38所示，注意变量的类型，同时注意初始值不能为0，否则没有星－三角起动效果。

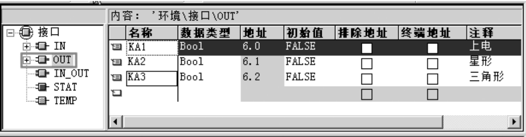



图5-37　接口“OUT”中，新建3个变量



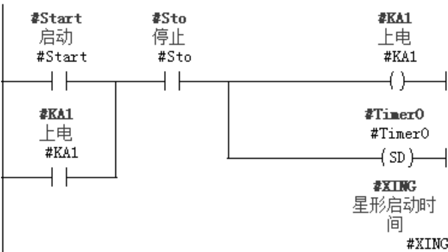

图5-38　在接口“STAT”中，新建2个静态变量

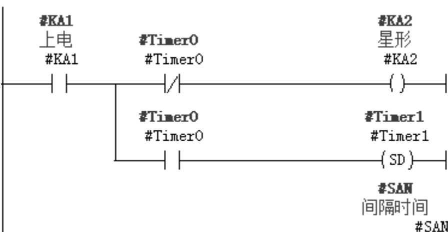
6）在FB1的程序编辑区编写程序，如图5-39所示。

7）在SIMATIC管理器界面中，双击“OB1”，打开主程序块“OB1”，如图5-40所示。

8）将功能“FB1”拖入程序段1，在FB1上方输入数据块DB1，如果这个数据块不存

在，那么 STEP 7 提示读者建立它，如图 5-41 所示。将整个项目下载到 PLC 中，即可实现“电动机星－三角起动控制”了。



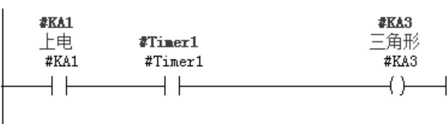




图 5-39　FB1 中的程序




图 5-40　打开主程序块

背景数据块 DB1 如图 5-42 所示。其地址（即第一列）含义如下：

0.0：表示 DB1.DBX0.0。

0.1：表示 DB1.DBX0.1。

2.0：表示 DB1.DBW2，并非表示 DB1.DBX2.0，“DB1.DBX2.0”是这个“字”的起始地址，这一点读者要特别注意。

4.0：表示 DB1.DBW4，并非表示 DB1.DBX4.0。

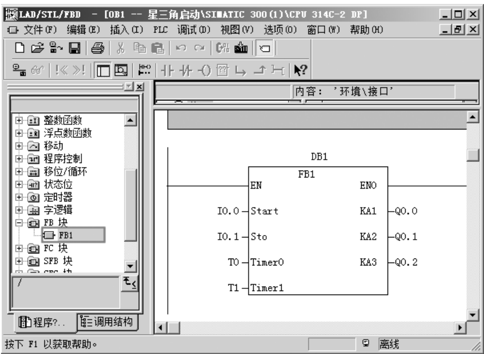



图 5-41　调用功能块

DB 参数 - DB1

数据块(A) 编辑(E) PLC(P) 调试(D) 查看(V) 窗口(W) 帮助(H)

DB1 -- 星三角启动\SIMATIC 300(1)\CPU 314C-2 DP

| | 地址 | 声明 | 名称 | 类型 | 初始值 | 实际值 | 备注 |
|---|---|---|---|---|---|---|---|
| 1 | 0.0 | in | Start | BOOL | FALSE | FALSE | 启动 |
| 2 | 0.1 | in | Sto | BOOL | FALSE | FALSE | 停止 |
| 3 | 2.0 | in | Timer0 | TIMER | T 0 | T 0 | |
| 4 | 4.0 | in | Timer1 | TIMER | T 0 | T 0 | |
| 5 | 6.0 | out | KA1 | BOOL | FALSE | FALSE | 上电 |
| 6 | 6.1 | out | KA2 | BOOL | FALSE | FALSE | 星形 |
| 7 | 6.2 | out | KA3 | BOOL | FALSE | FALSE | 三角形 |
| 8 | 8.0 | stat | XING | S5TIME | S5T#2S | S5T#2S | 星形启动时间 |
| 9 | 10.0 | stat | SAN | S5TIME | S5T#1S | S5T#1S | 间隔时间 |

离线

图 5-42　背景数据块 DB1

## 5.2.6　系统功能（SFC）及其应用

### 1. 系统功能（SFC）简介

系统功能（SFC）集成在 STEP 的“库”中，实际上是西门子公司编写的子程序，可以完成特定的功能。STEP 中有丰富的系统功能和功能块，供读者在编写程序时调用，这也是西门子 S7－300/400 PLC 功能强大的重要原因之所在。

### 2. 系统功能（SFC）应用

以下用一个例子说明系统功能的使用方法。

**【例 5-8】**用系统功能 SFC0 修改 CPU 314C－2DP 的系统时间。

**解：**

1）新建一个项目，本例为“时间”，选中“块”，双击“OB1”打开程序编辑器，如图 5-43 所示。

2）将系统功能拖入程序编辑区。展开“库”→“System Function Blocks”，先选中“SFC0”，再将“SFC0”拖入到程序编辑区，如图 5-44 所示。

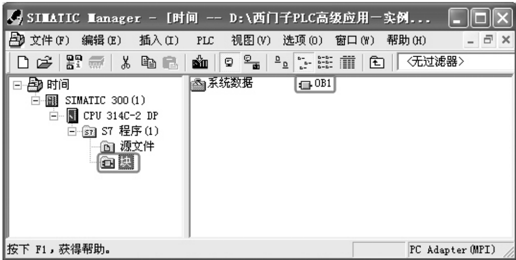



图 5-43　打开主程序块

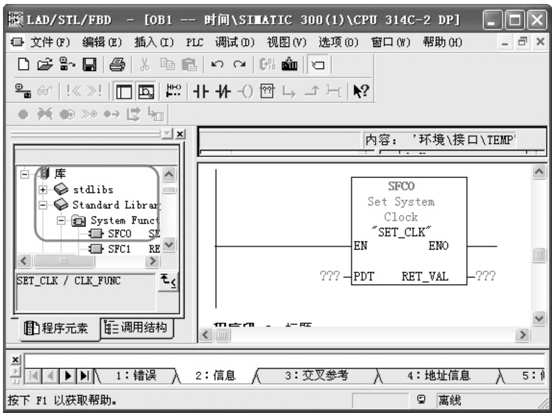



图 5-44　将系统功能拖入程序编辑区

3）编写如图 5-46 所示的梯形图程序。程序的作用是先将日期和时间合并，再用合并后的时间修改 CPU 的系统时间，修改后的系统时间是“2015-3－18－18:30:18.8”。

在 OB1 中，新建一个临时变量“SetTime”，如图 5-45 所示，注意其数据类型为“Date_And_Time”。如果数据类型不正确，则梯形图中变量“SetTime”是红色的，表示有错误。

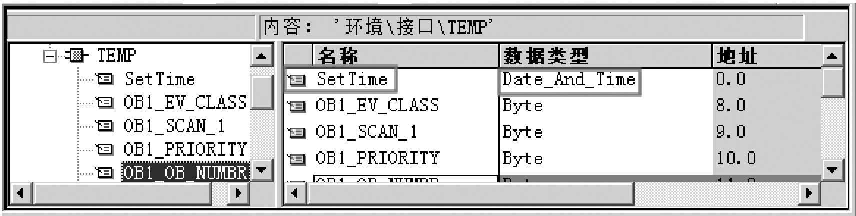
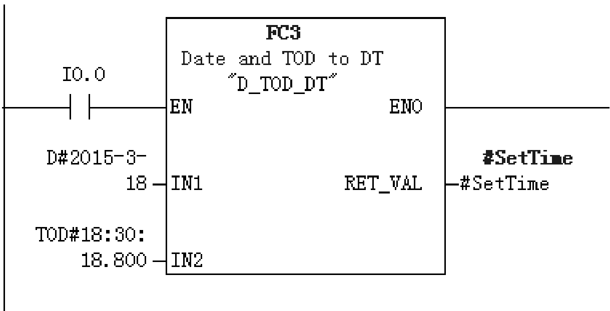



图 5-45　新建变量

设置 CPU 的时间还有一种比较简单的方法，不需要编写程序。先将 PG/PC 与 CPU 建立连接，再在 SIMATIC Manager 界面中，单击“PLC”→“诊断/设置（I）”→“设置时钟（T）”，如图 5-47 所示，弹出如图 5-48 所示的设置界面，单击“应用”按钮即可将需要设置的时间（可以修改计算机的时间）写入 CPU 中。

**程序段 1：** 日期和时间合并




**程序段 2：** 设置系统时间

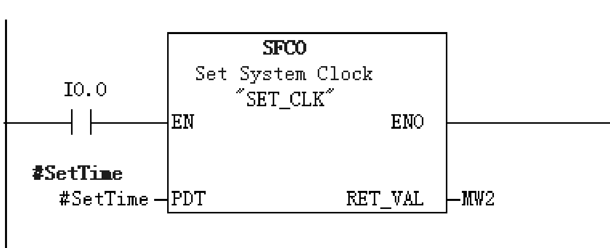



图 5-46　梯形图




图 5-47　梯形图　　　　图 5-48　梯形图

系统功能块（SFB）的使用方法和系统功能（SFC）的使用方法类似，只不过系统功能块要用到背景数据块。在后续章节有系统功能和系统功能块的具体应用，在此不再赘述。

# 5.3 多重背景

## 5.3.1 多重背景的简介

**1. 多重背景的概念**

当程序中有多个功能块时，如每个功能块对应一个背景数据块，程序中需要较多的背景数据块，这样在项目中就出现了大量的背景数据“碎片”，影响程序的执行效率。使用多重背景，可以将几个功能块，共用一个背景数据块，这样可以减少数据块的个数，提高程序的执行效率。

如图 5-49 所示是一个多重背景的结构实例。FB1 和 FB2 共用一个背景数据块 DB10，但增加了一个功能块 FB10 来调用作为“局部背景”的 FB1 和 FB2，而 FB1 和 FB2 的背景数据存放在 FB10 的背景数据块 DB10 中，如不使用多重背景，则需要两个背景数据块，使用多重背景后，则只需要一个背景数据块。

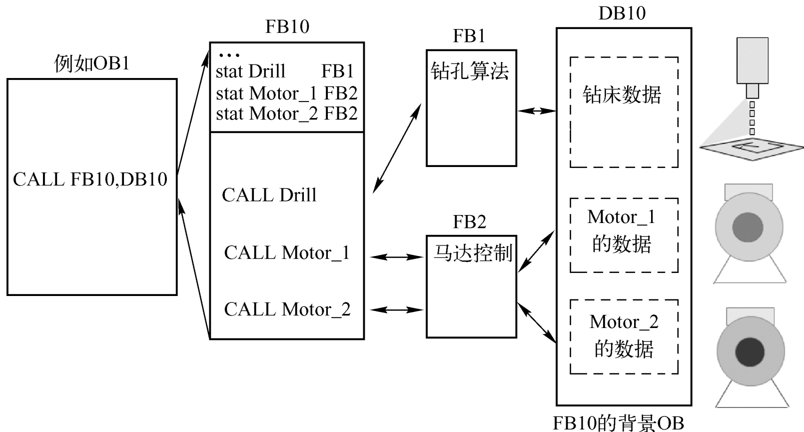



图 5-49 多重背景的结构

**2. 多重背景的优点**

1）多个实例只需要一个 DB。

2）在为各个实例创建“私有”数据区时，无需任何额外的管理工作。

3）多重背景模型使得“面向对象的编程风格”成为可能（通过“集合”的方式实现可重用性）。

## 5.3.2 多重背景的应用

以下用两个例子介绍多重背景的应用。

**【例 5-9】** 使用多重背景实现功能：电动机的起停控制和水位 AD 转换值大于 3000 时报警输出。

**解：**

1）新建项目和 3 个空的功能块，如图 5-50 所示，双击并打开 FB1，并在 FB1 中创建起停控制功能的程序，如图 5-51 所示。



图 5-50　新建项目和 3 个空的功能块

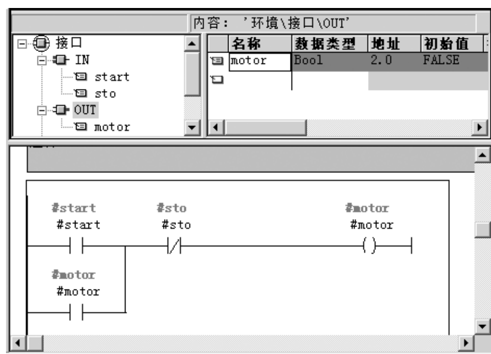


图 5-51　功能块 FB1

2）双击打开功能块 FB2，如图 5-52 所示，FB2 能实现当输入超过 3000 时报警的功能。

3）双击打开功能块 FB10，如图 5-53 所示，再展开静态变量“STAT”，并创建两个静态变量，静态变量“Qingting”的数据类型为“FB1”，静态变量“Baojing”的数据类型为“FB2”。FB10 中的梯形图如图 5-54 所示。

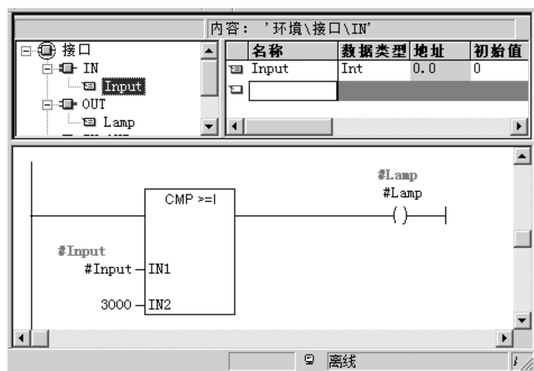


图 5-52　功能块 FB2



图 5-53　功能块 FB10

4）双击打开组织块 OB1，OB1 中的梯形图如图 5-55 所示。

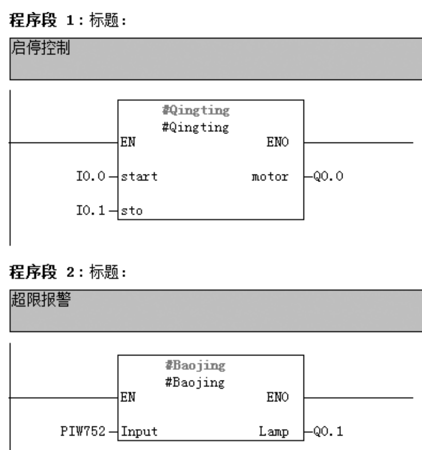



图 5-54　功能块 FB10 中的梯形图

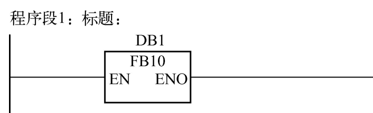



图 5-55　OB1 中的梯形图

当 PLC 的定时器不够用时，就要用到 IEC 定时器，IEC 定时器（如 SFB4）虽然可以多次调用，但需要消耗较多的数据块。而使用了多重背景则可减少 DB 的使用数量。

**【例 5-10】**编写程序实现，当 I0.0 闭合 2 s 后，Q0.0 线圈得电，当 I0.1 闭合 3 s 后，Q0.1 线圈得电，要求用 SFB4 定时器。

**解**：为节省 DB，就要使用多重背景。步骤如下：

1）先新建项目和 1 个空的功能块，双击并打开 FB1，并在静态变量“STAT”中创建“TON1”和“TON2”，其数据类型为“TON”，如图 5-56 所示。再在 FB1 中编写如图 5-57 所示的梯形图程序。

2）在 OB1 中编写如图 5-58 所示的梯形图程序。

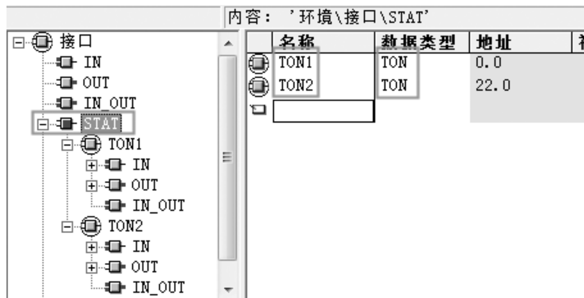
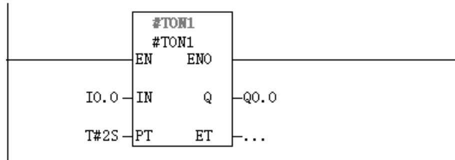



图 5-56　新建项目和空的功能块



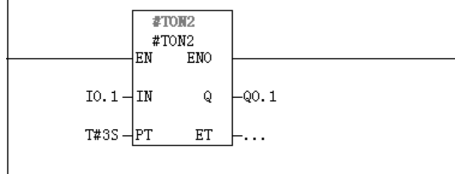




图 5-57　FB1 中的梯形图

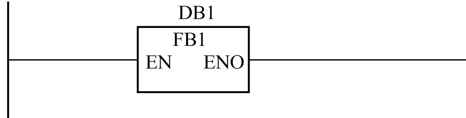



图 5-58　OB1 中的梯形图

## 5.4　库的应用

### 5.4.1　库的介绍

库用于存放 S7/M7 中可以多次使用的程序部件。这些程序部件可从已有的项目复制到库中，也可以直接在库中生成。库程序实际就是用户多次调用的程序块，安装了 STEP 7 后，软件中自动安装了标准库（Standard Library），标准库中有大量的程序块供编程者使用。编程者也可以把常用的程序，编写成程序块放在库中，作为库的程序部件。

## 5.4.2 库的应用介绍

**1. 标准库**

标准库（Standard Library）是西门子公司编写的可以实现特定功能的程序块，为编程者编写程序提供了极大的便利，在工程实践中，标准库极为常用。西门子的标准库如图 5-59 所示。

标准库包括如下功能块：

① PROFIenergy 相关的块：基于 PROFINET 现场总线实现自动生产线装备的能耗管理相关的块。

② 通信块（Communication Blocks）：用于通信的各种功能块，例如：用于 DP 通信的功能 FC1/FC2。

③ 组织块（Organization Blocks）：标准组织块（OB），例如：OB35。

④ S5-S7 转换块（S5-S7 Converting Blocks）：用于转换 STEP5 的程序块，目前已经较少使用了。

⑤ 系统功能块（System Function Blocks）：包括系统功能块（SFB）和系统功能（SFC）。

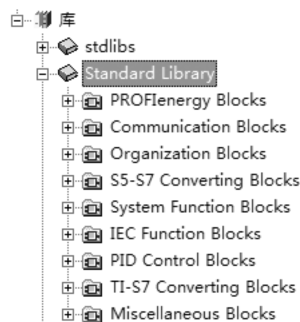



图 5-59 标准库

⑥ IEC 功能块（IEC Function Blocks）：用于 IEC 的功能块，例如：用于 IEC 的定时器和 IEC 的计数器等。

⑦ PID 控制块（PID Control Blocks）：用于 PID 控制块，例如：FB41。

⑧ TI-S7 转换块（TI-S7 Converting Blocks）：用于转换的块，如较常用的 FC105/FC106。

⑨ 其他块（Miscellaneous Blocks）：例如：用于时间标签的块。

标准库的调用方法在前述的章节中已经用到，在此不再赘述。

**2. 自建库**

用户可以把工作中常用的一些程序编写成功能或者功能块，添加到库中去，这样以后编写程序时就可以多次调用了，节省编程时间。以下用一个例子介绍自建库的实现过程。

**【例 5-11】** 将起停控制功能添加到库中。

**解：**

1）先创建一个起停控制的功能 FC110，输入参数、输出参数和梯形图程序如图 5-60 所示。

2）新建库。新建项目，选中“库”选项卡，新建的库名称为“My Lib”，单击“确定”按钮，如图 5-61 所示，新建完成一个空的库。

3）插入程序。在菜单栏，单击“插入”→“程序”→“S7 程序”，如图 5-62 所示。

4）将编写完成的功能 FC110 复制到库中，如图 5-63 所示，只要选中图中下面的“FC110”，按不住鼠标左键不放，拖拽到图中上部的空白处，松开鼠标左键，复制完成。

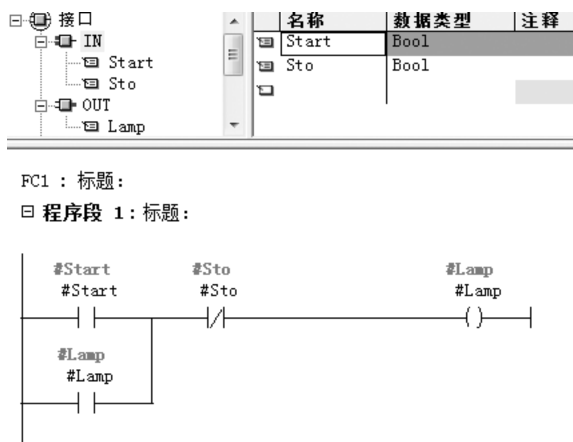






图 5-60　输入参数、输出参数和梯形图程序　　　　图 5-61　新建库

5）打开程序编辑中的“库”，库中有“My Lib”和“FC110”，如图 5-64 所示，说明起停控制的“功能”已经添加到自建的库“My Lib”中了。







图 5-62　插入程序　　　　图 5-63　复制功能 FC110 到库中

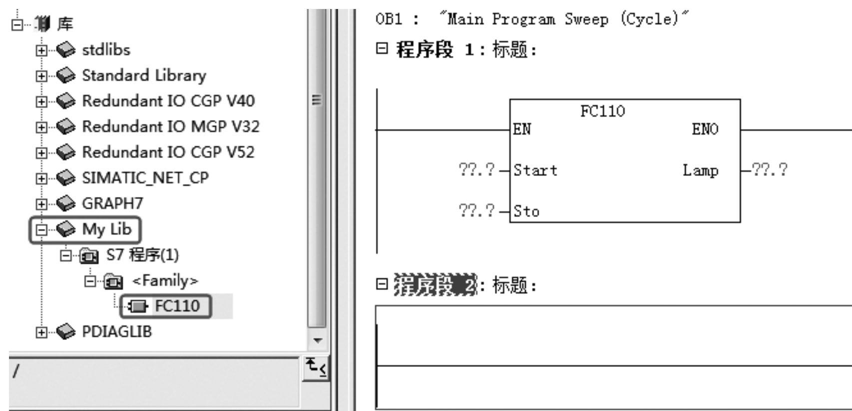



图 5-64　在库中调用新建的功能

## 5.5 组织块（OB）及其应用

组织块（OB）是操作系统与用户程序之间的接口。组织块由操作系统调用，控制循环中断驱动的程序执行、PLC 起动特性和错误处理。可以对组织块进行编程来确定 CPU 特性。

### 5.5.1 中断的概述

**1. 中断过程**

中断处理用来实现对特殊内部事件或外部事件的快速响应。CPU 检测到中断请求时，立即响应中断，调用中断源对应的中断程序（OB）。执行完中断程序后，返回被中断的程序。例如在执行主程序 OB1 块时，时间中断块 OB10 可以中断主程序块 OB1 正在执行的程序，转而执行中断程序块 OB10 中的程序，当中断程序块中的程序执行完成后，再转到主程序块 OB1 中，从断点处执行主程序。

中断源就是能向 PLC 发出中断请求的中断事件，例如日期时间中断、延时中断、循环中断和编程错误引起的中断。

**2. 中断的优先级**

执行一个组织块 OB 的调用可以中断另一个 OB 的执行。一个 OB 是否允许另一个 OB 中断取决于其优先级。OB 共有 29 个优先级，1 最低，29 最高。高优先级的 OB 可以中断低优先级的 OB。例如 OB10 的优先级是 2，而 OB1 的优先级是 1，所以 OB10 可以中断 OB1。背景 OB 的优先级最低。

优先级的顺序（后面的比前面的优先级高）：背景循环、主程序扫描循环、日期时间中断、时间延时中断、循环中断、硬件中断、多处理器中断、I/O 冗余错误、异步故障（OB80 ~ 87）、起动和 CPU 冗余，背景循环的优先级最低。

**3. 对中断的控制**

日期时间中断和延时中断有专用的允许处理中断和禁止中断的系统功能（SFC）。SFC 39“DIS_INT”用来禁止所有的中断、某些优先级范围的中断或指定的某个中断。SFC 40“EN_INT”用来激活（使能）新的中断和异步错误处理。如果用户希望忽略中断，可以下载一个只有块结束指令 BEU 的空 OB。

SFC 41“DIS_AIRT”延迟处理比当前优先级高的中断和异步错误。SFC 42“EN_AIRT”允许立即处理被 SFC 41 暂时禁止的中断和异步错误。

**4. 组织块的分类**

组织块只能由操作系统起动，它由变量声明表和用户编写的控制程序组成。

① 起动组织块 OB100 ~ OB102。

② 循环执行的组织块。

③ 定期执行的组织块。

④ 事件驱动的组织块。

延时中断、硬件中断、异步错误中断 OB80 ~ OB87，同步错误中断 OB121 和 OB122。

组织块的类型和优先级见表 5-3。

不是所有的中断组织块都能被 CPU 使用，不同类型的 CPU 可以调用的组织块一般不

同，例如有的型号 CPU 314C－2DP 的循环中断仅能调用组织块 OB35，而不能调用 OB30～OB34 和 OB36～OB38 组织块。

**表 5-3　组织块的类型和优先级**

| 中断类型 | 组　织　块 | 优先级（默认） | 启动事件 |
|---|---|---|---|
| 主程序扫描 | OB1 | 1 | 用于循环程序处理的组织块（OB1） |
| 时间中断 | OB10 到 OB17 | 2 | 时间中断组织块（OB10～OB17） |
| 延时中断 | OB20 | 3 | 延时中断组织块（OB20～OB23） |
| | OB21 | 4 | |
| | OB22 | 5 | |
| | OB23 | 6 | |
| 循环中断 | OB30 | 7 | 循环中断组织块（OB30～OB38） |
| | OB31 | 8 | |
| | OB32 | 9 | |
| | OB33 | 10 | |
| | OB34 | 11 | |
| | OB35 | 12 | |
| | OB36 | 13 | |
| | OB37 | 14 | |
| | OB38 | 15 | |
| 硬件中断 | OB40 | 16 | 硬件中断组织块（OB40～OB47） |
| | OB41 | 17 | |
| | OB42 | 18 | |
| | OB43 | 19 | |
| | OB44 | 20 | |
| | OB45 | 21 | |
| | OB46 | 22 | |
| | OB47 | 23 | |
| DPV1 中断 | OB 55 | 2 | 编程 DPV1 设备 |
| | OB 56 | 2 | |
| | OB 57 | 2 | |
| 多值计算中断 | OB60 多值计算 | 25 | 多值计算－多个 CPU 的同步操作 |
| 同步循环中断 | OB 61 | 25 | 组态 PROFIBUS DP 上的快速和等长过程响应时间 |
| | OB 62 | 25 | |
| | OB 63 | 25 | |
| | OB 64 | 25 | |
| 冗余错误 | OB70 I/O 冗余错误（仅在 H 系统中） | 25 | 错误处理组织块（OB70～OB87／OB121～OB122） |
| | OB72 CPU 冗余错误（仅在 H 系统中） | 28 | |

（续）

| 中断类型 | 组织块 | 优先级（默认） | 启动事件 |
|---|---|---|---|
| 异步错误 | OB80 时间错误 | 25，（但在起动程序中出现异步错误 OB，那么为28） | 错误处理组织块（OB70～OB87／OB121～OB122） |
| | OB81 电源错误 | | |
| | OB82 诊断错误 | | |
| | OB83 插入/删除模块中断 | | |
| | OB84 CPU 硬件故障 | | |
| 异步错误 | OB 85 程序周期错误 | | |
| | OB86 机架故障 | | |
| | OB87 通信错误 | | |
| 后台循环 | OB90 | 29 | 后台组织块（OB90） |
| 起动 | OB100 重起动，（热重起动） | 27 | 起动组织块（OB100/OB101/OB102） |
| | OB101 热重起动 | 27 | |
| | OB102 冷重起动 | 27 | |
| 同步错误 | OB121 编程错误 | 引起错误的 OB 的优先级 | 错误处理组织块（OB70～OB87／OB121～OB122） |
| | OB122 访问错误 | | |

## 5.5.2 主程序（OB1）

CPU 的操作系统循环执行 OB1。当操作系统完成起动后，将起动执行 OB1。在 OB1 中可以调用功能（FC）、系统功能（SFC）、功能块（FB）和系统功能块（SFB）。

执行 OB1 后，操作系统发送全局数据。重新起动 OB1 之前，操作系统将过程映像输出表写入输出模块中，更新过程映像输入表以及接受 CPU 的任何全局数据。

## 5.5.3 日期时钟中断组织块及其应用

CPU 可以使用的日期时间中断 OB 的个数与 CPU 的型号有关，例如 CPU314C－2DP 只能用 OB10。

**1. 指令简介**

日期时钟中断组织块可以在某一特定的日期和时间执行一次，也可以从设定的日期时间开始，周期性地重复执行，例如每分钟、每小时、每天，每年执行一次。可以用 SFC28～SFC31 设置、取消、激活和查询日期时间中断。SFC28～SFC31 的参数见表 5-4。

**表 5-4 SFC28～SFC31 的参数表**

| 参数 | 声明 | 数据类型 | 存储区间 | 参数说明 |
|---|---|---|---|---|
| OB_NR | INPUT | INT | I、Q、M、D、L、常数 | OB 的编号 |
| SDT | INPUT | DT | D、L、常数 | 起动日期和时间：将忽略指定的起动时间的秒和毫秒，并将其设置为 0 |

（续）

| 参　　数 | 声　　明 | 数据类型 | 存储区间 | 参数说明 |
|---|---|---|---|---|
| PERIOD | INPUT | WORD | I、Q、M、D、L、常数 | 从起动点 SDT 开始的周期：<br>W#16#0000 = 一次<br>W#16#0201 = 每分钟<br>W#16#0401 = 每小时<br>W#16#1001 = 每日<br>W#16#1202 = 每周<br>W#16#1401 = 每月<br>W#16#1801 = 每年<br>W#16#2001 = 月末 |
| RET_VAL | OUTPUT | INT | I、Q、M、D、L | 如果出错，则 RET_VAL 的实际参数将包含错误代码 |
| STATUS | OUTPUT | WORD | I、Q、M、D、L | 时间中断的状态 |

**2. 日期时钟中断组织块的应用**

以下通过例题说明日期时钟中断组织块的应用。

**【例 5-12】**从 2015 年 5 月 18 日 18 时 18 分起，每 1 小时中断一次，并将中断次数记录在一个存储器中。

**解：**一般有两种解法：

第一种解法比较简单，先打开 CPU 的属性界面，在“日期时钟中断”选项卡中，选择“激活”→“每小时”→“2015-5-18”→“18：18”，单击“确定”按钮，如图 5-65 所示。这个步骤的含义是：激活组织块 OB10 的中断功能，从 2015 年 5 月 18 日 18 时 18 分起，每 1 小时中断一次，再将组态完成的硬件下载到 CPU 中。

**【关键点】***初学者在使用此方法时，很容易忘记勾选“激活”或者不把组态的信息下载到 CPU 中去，请读者避免这样的失误。*




图 5-65　设置和激活日期时钟中断

打开 OB10，在程序编辑器中，输入程序如图 5-67 所示，运行的结果是从 2015 年 5 月 18 日 18 时 18 分起，每小时 MW2 中的数值增加 1，也就是记录了中断的次数。

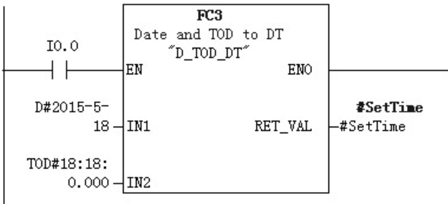
第二种解法，主程序在 OB1 中，如图 5-66 所示，中断程序在 OB10 中，如图 5-67 所示。

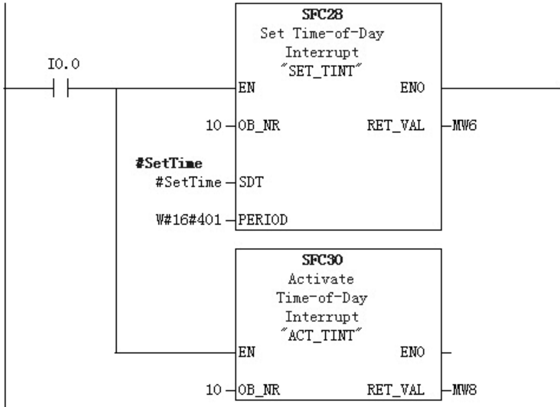
## 5.5.4　循环中断组织块及其应用

CPU 可以使用的循环中断 OB 的个数与 CPU 的型号有关。所谓循环中断就是经过一段固定的时间间隔中断用户程序。

**1. 循环中断指令**

循环中断组织块是很常用的，STEP 7 中有 9 个循环中断组织块（OB30 ~ OB38）。指令 SFC39 ~ SFC42 来激活循环中断、禁止循环中断、禁用报警中断和启用报警中断。指令 SFC39 ~ SFC42 的参数见表 5-5。



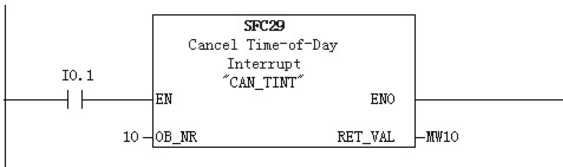




图 5-66　OB1 中的程序

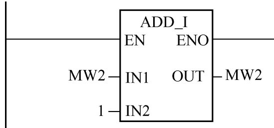



图 5-67　OB10 中的程序

**表 5-5　SFC39 ~ SFC42 的参数表**

| 参　　数 | 声　　明 | 数 据 类 型 | 存 储 区 间 | 参 数 说 明 |
|---|---|---|---|---|
| OB_NR | INPUT | INT | I、Q、M、D、L、常数 | OB 的编号 |
| MODE | INPUT | BYTE | I、Q、M、D、L、常数 | 指定禁用哪些中断和异步错误 |
| RET_VAL | OUTPUT | INT | I、Q、M、D、L | 如果出错，则 RET_VAL 的实际参数将包含错误代码 |

参数 MODE 指定禁用哪些中断和异步错误，含义比较复杂，MODE =0 表示激活所有的中断和异步错误，MODE =1 表示禁用所有新发生的和属于指定中断等级的事件，MODE =2 表示禁用所有新发生的指定中断。具体可参考相关手册。

**2. 循环中断组织块的应用**

**【例 5-13】** 每隔 100 ms，CPU 314C -2DP 采集一次通道 0 上的模拟量数据。

**解：** 很显然要使用循环组织块。解法如下：

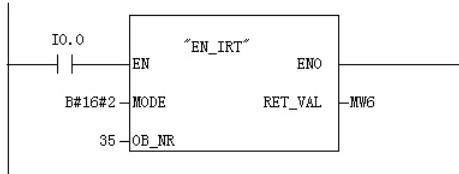
先打开 CPU 的属性界面，在“循环中断”选项卡中，将组织块 OB35 的执行时间定为“100 ms”，单击“确定”按钮，如图 5-68 所示。这个步骤的含义是：设置组织块 OB35 的循环中断时间是 100 ms。再将组态完成的硬件下载到 CPU 中。



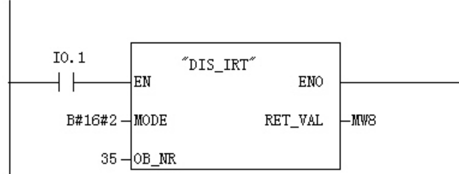

图 5-68　设置循环中断

打开 OB35，在程序编辑器中，输入程序如图 5-70 所示，运行的结果是每 100ms 将通道 0 的采集到模拟量转化成数字量送到 MW0 中。主程序在 OB1 中，如图 5-69 所示。有了主程序，就可以对 OB35 是否循环扫描中断进行控制了。

程序段 1：激活循环中断，组织块为OB35




程序段 2：取消循环中断



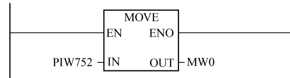




图 5-69　OB1 中的程序　　　　图 5-70　OB35 中的程序

### 5.5.5　硬件中断组织块及其应用

硬件中断组织块（OB40～OB47）用于快速响应信号模块（SM 输入/输出模块）、通信处理器（CP）和功能模块（FM）的信号变化。

硬件中断被模块触发后，操作系统将自动识别是哪一个槽的模块和模块中哪一个通道产生的硬件中断。硬件中断 OB 执行完后，将发送通道确认信号。

如果正在处理某一中断事件，又出现了同一模块同一通道产生的完全相同的中断事件，新的中断事件将丢失。

如果正在处理某一中断信号时，同一模块中其他通道或其他模块产生了中断事件，当前已激活的硬件中断执行完后，再处理暂存的中断。

以下通过一个例子说明硬件中断组织块的使用方法。

【例 5-14】编写一段指令记录用户使用 I3.0 和 I3.1 按钮的次数，做成一个简单的“黑匣子”。

**解**：系统的硬件为 CPU 314C-2DP 和输入信号模块 SM321（Interrupt，带硬件中断功能）。先进行硬件组态，如图 5-71 所示，很明显，信号输入模块的输入地址为“IB3”和“IB4”。双击“SM321 DI16XDC24V，Interrupt”，弹出信号模块的属性界面，如图 5-72 所示。在“输入选项卡”中，勾选“硬件中断”和“上升沿硬件中断发生器”（实际就是对 I3.0 和 I3.1 有效），最后单击“确定”按钮。

在组织块 OB40 中编写程序，如图 5-73 所示，每次按下按钮，调用 OB40 中的程序一次，MW0 中的数值加 1，也就是记录了使用按钮的次数。

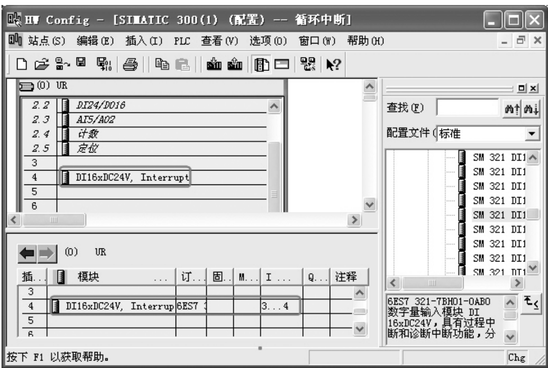



图 5-71 硬件组态界面




图 5-72 信号模块的属性界面

【关键点】选用的输入模块“DI16XDC24V，Interrupt”必须具有硬件中断功能。

这个例子也可以用 SFC39 和 SFC40 来取消和激活中断。

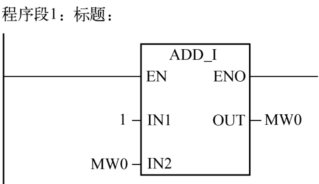



图 5-73 OB40 中的程序

## 5.5.6 错误处理组织块及其应用

### 1. 错误处理概述

S7-300/400 PLC 具有很强的错误（或称故障）检测和处理能力。PLC 内部的功能性错误或编程错误，而不是外部设备的故障。CPU 检测到错误后，操作系统调用对应的组织块，用户可以在组织块中编程，对发生的错误采取相应的措施。对于大多数错误，如果没有给组织块编程，出现错误时，CPU 将进入 STOP 模式。

### 2. 错误的分类

被 S7 CPU 检测到并且用户可以通过组织块对其进行处理的错误分为两个基本类型：

（1）异步错误

是与 PLC 的硬件或操作系统密切相关的错误，与程序执行无关，后果严重。异步错误 OB 具有最高等级的优先级，其他 OB 不能中断它们。同时有多个相同优先级的异步错误 OB 出现，将按出现的顺序处理。

当系统程序可以检测下列错误：不正确的 CPU 功能、系统程序执行中的错误、用户程序中的错误和 I/O 中的错误时，根据错误类型的不同，CPU 设置为进入 STOP 模式或调用一个错误处理组织块（OB）。

当 CPU 检测到错误时，会调用适当的组织块，见表 5-6。如果没有相应的错误处理 OB，CPU 将进入 STOP 模式。用户可以在错误处理 OB 中编写如何处理这种错误的程序，以减小或消除错误的影响。

**表 5-6　错误处理组织块**

| OB 号 | 错误类型 | 优先级 |
|---|---|---|
| OB70 | I/O 冗余错误（仅 H 系列 CPU） | 25 |
| OB72 | CPU 冗余错误（仅 H 系列 CPU） | 28 |
| OB73 | 通信冗余错误（仅 H 系列 CPU） | 35 |
| OB80 | 时间错误 | 26 |
| OB81 | 电源错误 | 26/28 |
| OB82 | 诊断中断 | |
| OB83 | 插入/取出模块中断 | |
| OB84 | CPU 硬件故障 | |
| OB85 | 优先级错误 | |
| OB86 | 机架故障或分布式 I/O 的站故障 | |
| OB87 | 通信错误 | |
| OB121 | 编程错误 | 引起错误的 OB 的优先级 |
| OB122 | I/O 访问错误 | |

为避免发生某种错误时 CPU 进入停机，可以在 CPU 中建立一个对应的空的组织块。用户可以利用 OB 中的变量声明表提供的信息来判别错误的类型。

（2）同步错误（OB121 和 OB122）

是与程序执行有关的错误，其 OB 的优先级与出现错误时被中断的块的优先级相同，即同步错误 OB 中的程序可以访问块被中断时累加器和状态寄存器中的内容。对错误进行处理后，可以将处理结果返回被中断的块。

**3. 时间错误处理组织块（OB80）**

OB 执行时出现时间故障 S7-300 CPU 的操作系统调用 OB80。这样的故障包括循环时间超出、执行 OB 时应答故障、向前移动时间以致于跃过了 OB 的起动的时间以及 CLR 后恢复 RUN 方式。

如果当循环中断 OB 仍在执行前一次调用时，该 OB 块的起动事件发生，操作系统调用 OB80。如果 OB80 未编程，CPU 变为 STOP 方式，可以使用 SFC39 ~ SFC42 封锁或延时，或再使用时间故障 OB。

如果在同一个扫描周期中由于扫描时间超出 OB80 被调用两次，CPU 就变为 STOP 方式，可以通过在程序中适当的位置调用 SFC43 “RE_TRIGR” 来避免这种情况。OB80 的变量声明表见表 5-7。

**表 5-7　OB80 的变量声明表**

| 变量 | 类型 | 描述 |
|---|---|---|
| OB80_EV_CLASS | BYTE | 事件级别和标识：B#16#35 |
| OB80_FLT_ID | BYTE | 故障代码 |

（续）

| 变　　量 | 类　　型 | 描　　述 |
|---|---|---|
| OB80_PRIORITY | BYTE | 优先级：在 RUN 方式时 OB80 以优先级 26 运行，OB 请求缓冲区溢出时以优先级 28 运行 |
| OB80_OB_NUMBR | BYTE | OB 号 |
| OB80_RESERVED_1 | BYTE | 保留 |
| OB80_RESERVED_2 | BYTE | 保留 |
| OB80_ERROR_INFO | WORD | 故障信息：根据故障代码 |
| OB80_ERR_EV_CLASS | BYTE | 引起故障的起动事件的事件级别 |
| OB80_ERR_EV_NUM | BYTE | 引起故障的起动事件的事件号 |
| OB80_OB_PRIORITY | BYTE | 故障信息：根据故障代码 |
| OB80_OB_NUM | BYTE | 故障信息：根据故障代码 |
| OB80_DATE_TIME | DATE_AND_TIME | OB 被调用时的日期和时间 4 |

以下的指令表程序，在组织块 OB80 中，其功能是：当 CPU 发生如循环时间超出等故障时，执行 OB80 中的程序（如下程序），使得 M0.0 得电。如果 M0.0 和 HMI 关联，当 CPU 发生时间处理故障时，可以发出报警信息，显示在 HMI 上。

```
      SET
      SAVE
       =      L      20.1
      A       M      0.0
      NOT
      JCN     MOO1
      SET
       =      M      0.0
MOO1:CLR
      A       L      20.1
      SAVE
      BE
```

**4. 电源故障处理组织块（OB81）**

与电源（仅对 S7－400 PLC）或后备电池有关的故障事件发生时，S7－400 CPU 的操作系统调用 OB81。如果 OB81 未编程，CPU 不会转换为 STOP 方式。可以使用 SFC39～SFC42 来禁用、延时或再使用电源故障（OB81）。OB81 的变量声明表见表 5-8。

**表 5-8　OB81 的变量声明表**

| 变　　量 | 类　　型 | 描　　述 |
|---|---|---|
| OB81_EV_CLASS | BYTE | 事件级别和标识：B#16#38，离去事件；B#16#39，到来事件 |
| OB81_FLT_ID | BYTE | 故障代码 |
| OB81_PRIORITY | BYTE | 优先级：可通过 STEP 7 选择（硬件组态） |
| OB81_OB_NUMBR | BYTE | OB 号 |
| OB81_RESERVED_1 | BYTE | 保留 |
| OB81_RESERVED_2 | BYTE | 保留 |
| OB81_MDL_ADDR | INT | 位 0 至 2：机架号；位 3：0 = 备用 CPU，1 = 主站 CPU；位 4 至 7：1111 |
| OB81_RESERVED_3 | BYTE | 仅与部分故障代码有关 |
| OB81_RESERVED_4 | BYTE | |
| OB81_RESERVED_5 | BYTE | |
| OB81_RESERVED_6 | BYTE | |
| OB81_DATE_TIME | DATE_AND_TIME | OB 被调用时的日期和时间 |

以下的指令表程序，在组织块 OB81 中，其功能是当 CPU 发生电源故障时，执行 OB81 中的程序（如下程序）。如果检测到 OB81_EV_CLASS 等于 16#39，表示事件到来，使 M0.1 =1，而当检测到 OB81_EV_CLASS 等于 16#38，表示事件离开，使 M0.1 =0。如果 M0.1 和 HMI 关联，可以在 HMI 上显示电源故障的到来和离开。

```
        SET
        =          L    20.1
        L          #OB81_EV_CLASS
        L          B#16#39
        ==I
        JCN        MOO1
        SET
        =          M    0.1
        JU         MOO2
MOO1:   L          #OB81_EV_CLASS
        L          B#16#38
        ==I
        JCN        MOO2
        CLR
        =          M    0.1
MOO2:   CLR
        A          L    20.1
        SAVE
        BE
```

**5. 诊断中断处理组织块（OB82）**

如果模块具有诊断能力又使能了诊断中断，当它检测到错误时，它输出一个诊断中断请求给 CPU，以及错误消失时，操作系统都会调用 OB82。当一个诊断中断被触发时，有故障的模块自动地在诊断中断 OB 的起动信息和诊断缓冲区中存入 4 个字节的诊断数据和模块的起始地址。可以用 SFC39 ~ SFC42 来禁用、延时或再使用诊断中断（OB82），表 5-9 描述了诊断中断 OB82 的临时变量。

**表 5-9　OB82 的变量声明表**

| 变　　量 | 类　　型 | 描　　述 |
|---|---|---|
| OB82_EV_CLASS | BYTE | 事件级别和标识：B#16#38，离去事件；B#16#39，到来事件 |
| OB82_FLT_ID | BYTE | 故障代码 |
| OB82_PRIORITY | BYTE | 优先级：可通过 SETP 7 选择（硬件组态） |
| OB82_OB_NUMBR | BYTE | OB 号 |
| OB82_RESERVED_1 | BYTE | 备用 |
| OB82_IO_FLAG | BYTE | 输入模板：B#16#54；输出模板：B#16#55 |
| OB82_MDL_ADDR | WORD | 故障发生处模板的逻辑起始地址 |
| OB82_MDL_DEFECT | BOOL | 模板故障 |
| OB82_INT_FAULT | BOOL | 内部故障 |
| OB82_EXT_FAULT | BOOL | 外部故障 |
| OB82_PNT_INFO | BOOL | 通道故障 |
| OB82_EXT_VOLTAGE | BOOL | 外部电压故障 |
| OB82_FLD_CONNCTR | BOOL | 前连接器未插入 |
| OB82_NO_CONFIG | BOOL | 模板未组态 |

（续）

| 变　　量 | 类　　型 | 描　　述 |
|---|---|---|
| OB82_CONFIG_ERR | BOOL | 模板参数不正确 |
| OB82_MDL_TYPE | BYTE | 位 0～3：模板级别；位 4：通道信息存在；位 5：用户信息存在；位 6：来自替代的诊断中断；位 7：备用 |
| OB82_SUB_MDL_ERR | BOOL | 子模板丢失或有故障 |
| OB82_COMM_FAULT | BOOL | 通信问题 |
| OB82_MDL_STOP | BOOL | 操作方式（0：RUN，1：STOP） |
| OB82_WTCH_DOG_FLT | BOOL | 看门狗定时器响应 |
| OB82_RESERVED_2 | BOOL | 备用 |
| OB82_RACK_FLT | BOOL | 扩展机架故障 |
| OB82_PROC_FLT | BOOL | 处理器故障 |
| OB82_EPROM_FLT | BOOL | EPROM 故障 |
| OB82_RAM_FLT | BOOL | RAM 故障 |
| OB82_ADU_FLT | BOOL | ADC/DAC 故障 |
| OB82_FUSE_FLT | BOOL | 熔断器熔断 |
| OB82_HW_INTR_FLT | BOOL | 硬件中断丢失 |
| OB82_RESERVED_3 | BOOL | 备用 |
| OB82_DATE_TIME | DATE_AND_TIME | OB 被调用时的日期和时间 |

**【例 5-15】**控制系统为 S7－300 CPU 和 SM331，当传感器采集的电流大于 16mA 时报警，小于此值时停止报警，用 OB82 实现此功能。

**解：**

（1）硬件组态

要使用 OB82 诊断故障，SM331 模块必须有诊断功能。硬件组态过程如下：

系统硬件组态如图 5-74 所示，双击“AI8x12Bit”，弹出如图 5-75 所示的界面，配置 SM331 模块的“输入（Inputs）”选项，选择 0－1 通道组为 4 线制电流（4DMU），其他通道组为“取消激活”，并注意模块的量程卡要与设置的相同。选中“启用（Enable）”框中的“诊断中断（Diagnostic Interrupt）”选项，选中“诊断（Diagnostics）”选项中的 0－1 通道组中的“组诊断（Group Diagnostics）”选项，并在“上限框”中输入 16mA，最后单击“确定”按钮。

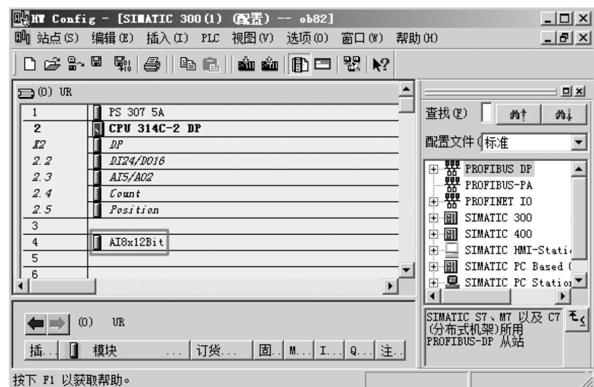



图 5-74　硬件组态




图 5-75　硬件组态

（2）编写梯形图程序

在编写梯形图程序之前，先要理解 OB82 的两个变量的含义。

OB82_EV_CLASS 是事件级别和标识（见表 5-9），当其为 B#16#38（56）时，表示离去事件；当其为 B#16#39（57）时，表示到来事件。对应本例，就是当传感器采集到的电流大于 16mA 时，表示事件到来，而当传感器采集到的电流小于 16 mA 时，表示事件离开。

OB82_MDL_ADDR 就是故障发生处模板的逻辑起始地址，就是模拟量模块的通道起始地址。

编写梯形图程序如图 5-76 所示。

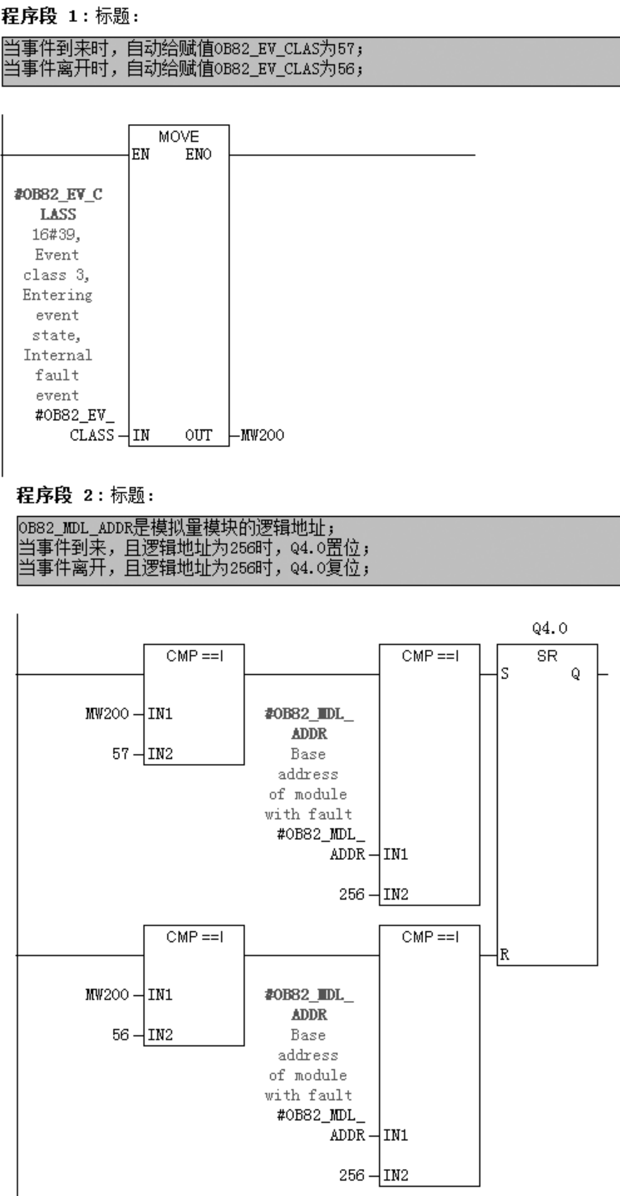



图 5-76　梯形图

（3）运行仿真

在仿真器的 PIW256 中输入大于 22119（此数字量对应 16mA 的模拟量），则 Q4.0 置位，而在仿真器的 PIW256 中输入小于 22119（此数字量对应 16mA 的模拟量），则 Q4.0 复位，但实际上，以上结果不会出现。正确的操作步骤是，先把完整的程序下载到仿真器中，将仿真器置于“RUN”状态，在工具栏中单击“Execute”（执行）→“Trigger Error OB”（触发器）→“Diagostic Interrupt”（诊断中断），如图 5-77 所示，打开诊断中断界面。

如图 5-78 所示，在“Module address”（模块地址）中输入故障模块的逻辑地址（即模拟量的通道号）“piw256”，任意选择一个可能的故障，本例选择“Channel fault”（通道故障），单击“Apply”（应用）按钮，仿真器中的 Q4.0 亮，如图 5-79 所示。同理，当去掉所有故障选项，再单击“Apply”（应用）按钮，仿真器中的 Q4.0 灭。

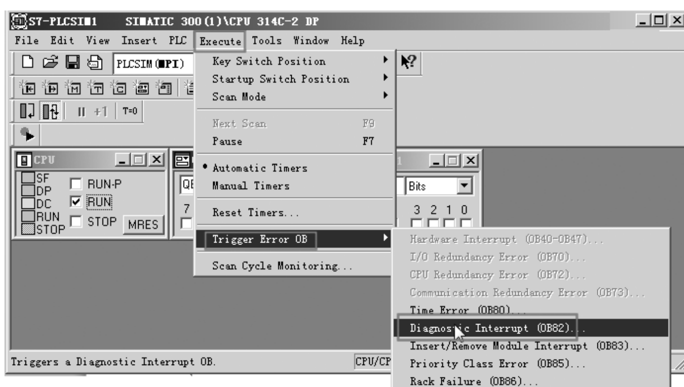



图 5-77 打开“诊断中断”界面

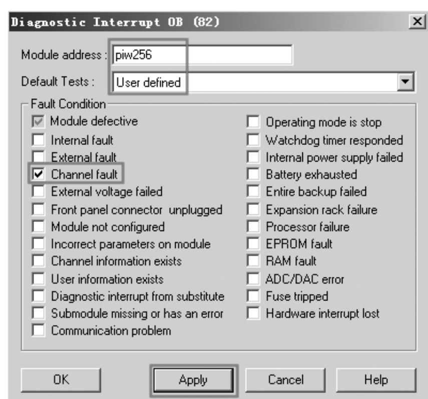



图 5-78 设置故障




图 5-79 故障显示

**【关键点】** 要利用 OB82 编写故障诊断的程序相对较难，关键要掌握组织块 OB82 的各项参数的含义。

此题用指令表编写更加简洁，如下所示：

```
SET
SAVE
```

```
        =       L    20.1
        L       #OB82_EV_CLASS
        L       B#16#39
        ==I
        JCN     MOO1
        SET
        =       Q    4.0
        JU      MOO2
MOO1:L          #OB82_EV_CLASS
        L       B#16#38
        ==I
        JCN     MOO2
        CLR
        =       Q    4.0
MOO2:CLR
        A       L    20.1
        SAVE
        BE
```

**6. 插入/拔出模块中断组织块（OB83）**

当组态的模块插入/拔出后或在 SETP 7 下修改了模块的参数并在“RUN”状态下把所做的修改下载到 CPU 后，CPU 操作系统调用 OB83。

在“RUN”“STOP”和“STARTUP”状态时，每次组态的模块插入或拔出就产生一个插入/拔出中断（电源模块、CPU、适配模块和 IM 模块不能在这种状态下拔出）。

OB83 的变量声明见表 5-10。

**表 5-10 OB83 的变量声明表**

| 变　　量 | 类　　型 | 描　　述 |
|---|---|---|
| OB83_EV_CLASS | BYTE | 事件级别和标识：B#16#32，模块参数赋值结束；B#16#33，模块参数赋值起动；B#16#38，模块插入；B#16#39，模块拔出或无反应，或参数赋值结束 |
| OB83_FLT_ID | BYTE | 故障代码 |
| OB83_PRIORITY | BYTE | 优先级，可通过 STEP 7 选择（硬件组态） |
| OB83_OB_NUMBR | BYTE | OB 号 |
| OB83_RESERVED_1 | BYTE | 块模块或接口模块标识 |
| OB83_MDL_ID | BYTE | 范围：B#16#54，外设输入（PI）；B#16#55，外设输出（PQ） |
| OB83_MDL_ADDR | WORD | 有关模块的逻辑起始地址 |
| OB83_RACK_NUM | WORD | B#16#A0，接口模块号；B#16#C4，机架号或 DP 站号（低字节）或 DP 主站系统 ID（高字节） |
| OB83_MDL_TYPE | WORD | 有关模块的模块类型 |
| OB83_DATE_TIME | DATE_AND_TIME | OB 被调用时的日期和时间 |

以下的指令表程序在组织块 OB83 中，其功能是插拔事件发生时，执行 OB83 中的程序（如下程序）。如果检测到 OB83_EV_CLASS 等于 16#39，表示事件到来，使 M0.2＝1；而当检测到 OB83_EV_CLASS 等于 16#38，表示事件离开，使 M0.2＝0。如果 M0.2 和 HMI 关联，可以在 HMI 上显示模块插拔事件的发生情况，即插拔事件的到来或者离开。

```
        SET
        SAVE
        =         L    20.1
        L         #OB83_EV_CLASS
        L         B#16#39
        ==I
        JCN       MOO1
        SET
        =         M    0.2
        JU        MOO2
MOO1:   L         #OB83_EV_CLASS
        L         B#16#38
        ==I
        JCN       MOO2
        CLR
        =         Q    0.2
MOO2:   CLR
        A         L    20.1
        SAVE
        BE
```

**7. CPU 硬件故障处理组织块（OB84）**

当 CPU 检测到 MPI 网络的接口故障、通信总线的接口故障或分布式 I/O 网卡的接口故障时，操作系统调用 OB84。故障消除时也会调用该 OB 块。OB84 的变量声明见表 5-11。

**表 5-11　OB84 的变量声明表**

| 变　量 | 类　型 | 描　述 |
|---|---|---|
| OB84_EV_CLASS | BYTE | 事件级别和标识：B#16#38，离去事件；B#16#39，到来事件 |
| OB84_FLT_ID | BYTE | 故障代码 |
| OB84_PRIORITY | BYTE | 优先级，可通过 STEP 7 选择（硬件组态） |
| OB84_OB_NUMBR | BYTE | OB 号 |
| OB84_RESERVED_1 | BYTE | 备用 |
| OB84_RESERVED_2 | BYTE | 备用 |
| OB84_RESERVED_3 | WORD | 备用 |
| OB84_RESERVED_4 | DWORD | 备用 |
| OB84_DATE_TIME | DATE_AND_TIME | OB 被调用时的日期和时间 |

以下的指令表程序在组织块 OB84 中，其功能是 CPU 硬件故障事件发生时，执行 OB84 中的程序（如下程序）。如果检测到 OB84_EV_CLASS 等于 16#39，表示事件到来，使 M0.3 =1；而当检测到 OB84_EV_CLASS 等于 16#38，表示事件离开，使 M0.3 =0。如果 M0.3 和 HMI 关联，可以在 HMI 上显示 CPU 硬件故障事件发生的情况，即硬件故障事件的到来或者离开。

```
        SET
        SAVE
        =         L    20.1
        L         #OB84_EV_CLASS
        L         B#16#39
        ==I
        JCN       MOO1
        SET
```

```
        =       M     0.3
        JU      MOO2
MOO1:L          #OB84_EV_CLASS
        L       B#16#38
        ==I
        JCN     MOO2
        CLR
        =       Q     0.3
MOO2:CLR
        A       L     20.1
        SAVE
        BE
```

**8. 优先级错误处理组织块（OB85）**

当以下情况发生时，将会触发优先级错误中断：

① 产生了一个中断事件，但是对应的 OB 块没有下载到 CPU。

② 访问一个系统功能块的背景数据块时出错。

③ 刷新过程映像表时，I/O 访问出错，模块不存在或有故障。

OB85 的变量声明见表 5-12。

**表 5-12　OB85 的变量声明表**

| 变　　量 | 类　　型 | 描　　述 |
|---|---|---|
| OB85_EV_CLASS | BYTE | 事件级别和标识 |
| OB85_FLT_ID | BYTE | 故障代码 |
| OB85_PRIORITY | BYTE | 优先级，可通过 STEP 7 选择（硬件组态） |
| OB85_OB_NUMBR | BYTE | OB 号 |
| OB85_RESERVED_1 | BYTE | 备用 |
| OB85_RESERVED_2 | BYTE | 备用 |
| OB85_RESERVED_3 | INT | 备用 |
| OB85_ERR_EV_CLASS | BYTE | 引起故障的事件级别 |
| OB85_ERR_EV_NUM | BYTE | 引起故障的事件号码 |
| OB85_OB_PRIOR | BYTE | 当故障发生时被激活 OB 的优先级 |
| OB85_OB_NUM | BYTE | 当故障发生时被激活 OB 的号码 |
| OB85_DATE_TIME | DATE_AND_TIME | OB 被调用时的日期和时间 |

以下的指令表程序在组织块 OB85 中，其功能是：当发生优先级错误时，执行 OB85 中的程序（如下程序），使得 M0.4 得电。如果 M0.4 和 HMI 关联，当发生通信故障时，可以发出报警信息，显示在 HMI 上。同时把错误类型存入 MW12 中，把错误编号存入 MW10 中。

```
        SET
        SAVE
        =       L     20.1
        A       M     0.4
        NOT
        JCN     MOO1
        SET
        =       M     0.4
MOO1:L          #OB85_FLT_ID
        AD      DW#16#FF
        T       MW    12
```

```
L        #OB85_OB_NUM
AD       DW#16#FF
T        MW      10
CLR
A        L      20.1
SAVE
BE
```

**9. 机架故障组织块（OB86）**

当以下情况发生时，CPU 的操作系统调用 OB86：

① 中央扩展单元（不带 S7－300）的故障已经删除（事件进入和退出状态）。

② DP 主站系统的故障已删除（事件进入和退出状态）。

③ 使用分布式 I/O（PROFIBUS DP 或 PROFINET IO）时，检测到站故障（事件进入和退出状态）。

④ 使用分布式 I/O（PROFIBUS DP 或 PROFINET IO）时，禁用了一个带 SFC 12"D_ACT_DP"和设置 MODE＝4 的站。

⑤ 使用分布式 I/O（PROFIBUS DP 或 PROFINET IO）时，启用了一个带 SFC 12"D_ACT_DP"和设置 MODE＝3 的站。

OB86 的变量声明见表 5-13。

**表 5-13　OB86 的变量声明表**

| 变　　量 | 类　　型 | 描　　述 |
|---|---|---|
| OB86_EV_CLASS | BYTE | 事件级别和标识：B#16#38，离去事件；B#16#39，到来事件 |
| OB86_FLT_ID | BYTE | 故障代码 |
| OB86_PRIORITY | BYTE | 优先级，可通过 STEP 7 选择（硬件组态） |
| OB86_OB_NUMBR | BYTE | OB 号 |
| OB86_RESERVED_1 | BYTE | 备用 |
| OB86_RESERVED_2 | BYTE | 备用 |
| OB86_MDL_ADDR | WORD | 根据故障代码 |
| OB86_Z23 | DWORD | 根据故障代码 |
| OB86_DATE_TIME | DATE_AND_TIME | OB 被调用时的日期和时间 |

以下程序在组织块 OB86 中，当机架故障到来或者离开时都会执行 OB86 中的程序。

```
A(
A(
A(
L        #OB86_EV_CLASS
T        MB       0
SET
SAVE
CLR
A        BR
)
JNB      _001
L        #OB86_FLT_ID
T        MB       1
SET
```

```
        SAVE
        CLR
_001:   A       BR
        )
        JNB     _002
        L       #OB86_MDL_ADDR
        T       MW        2
        SET
        SAVE
        CLR
_002:   A       BR
        )
        JNB     _003
        L       #OB86_Z23
        T       MD        4
_003:   NOP     0
```

① MB0 存入的是机架故障事件是否到来的信息（OB86_EV_CLASS），当 MB0 = 16#39 表示故障到来，当 MB0 = 16#38 表示故障离开。

② MB1 保存的是故障代码（OB86_FLT_ID），这个代码可与后续信息组合；MW2（OB86_MDL_ADDR）中存入是取决于错误代码，当 MB1 = 16#C4 时，MW2 中存入是 DP 主站的逻辑基址。

③ 当 MB1 = 16#C4 和 MB0 = 16#39/38 时，OB86_Z23（双字）含义如下：

位 0 ~ 7：DP 站编号；位 8 ~ 15：DP 主站系统 ID；位 16 ~ 30：S7 从站的逻辑基址或者标准 DP 从站的诊断地址；位 31：I/O 标识符。

如图 5-80 所示的变量监控了机架故障的信息，MB0 = 16#39 表示故障到来，MB1 = 16#C4，MW2 = 16#0FFF 表示逻辑地址；MD = 16#0FFF0103，分三部分解释其含义，16#0FFF 表示逻辑地址，16#01 表示主站地址，16#03 表示故障从站地址。

VAT_1 -- @OB86_2\SIMATIC 400(1)\CPU 412-1\S7 程序(1) ON...

| | 地址 | 符号 | 显示格式 | 状态值 | 修改数值 |
|---|---|---|---|---|---|
| 1 | //OB86_EV_CLASS | | | | |
| 2 | MB 0 | | HEX | B#16#39 | |
| 3 | //OB86_FLT_ID | | | | |
| 4 | MB 1 | | HEX | B#16#C4 | |
| 5 | //OB86_MDL_ADDR | | | | |
| 6 | MW 2 | | HEX | W#16#0FFF | |
| 7 | //OB86_Z23 | | | | |
| 8 | MD 4 | | HEX | DW#16#0FFF0103 | |
| 9 | | | | | |

图 5-80　变量表监控

**10. 通信错误组织块（OB87）**

在使用通信功能块或全局数据（GD）通信进行数据交换时，如果出现下列通信错误，操作系统将调用 OB87：

① 接收全局数据时，检测到不正确的帧标识符（ID）。

② 全局数据通信的状态信息数据块不存在或太短。

③ 接收到非法的全局数据包编号。

如果用于全局数据通信状态信息的数据块丢失，需要用 OB87 生成该数据块将它下载到 CPU。可以使用 SFC39 ~42 封锁或延时并使能通信错误 OB。

OB87 的变量声明见表 5-14。

**表 5-14　OB87 的变量声明表**

| 变　　量 | 类　　型 | 描　　述 |
|---|---|---|
| OB87_EV_CLASS | BYTE | 事件级别和标识 |
| OB87_FLT_ID | BYTE | 故障代码 |
| OB87_PRIORITY | BYTE | 优先级，可通过 SETP 7 选择（硬件组态） |
| OB87_OB_NUMBR | BYTE | OB 号 |
| OB87_RESERVED_1 | BYTE | 备用 |
| OB87_RESERVED_2 | BYTE | 备用 |
| OB87_RESERVED_3 | WORD | 根据故障代码 |
| OB87_RESERVED_4 | DWORD | 根据故障代码 |
| OB87_DATE_TIME | DATE_AND_TIME | OB 被调用时的日期和时间 |

以下的指令表程序在组织块 OB87 中，其功能是：当发生通信故障时，执行 OB87 中的程序（如下程序），使得 M0.5 得电。如果 M0.5 和 HMI 关联，当发生通信故障时，可以发出通信报警信息，显示在 HMI 上。

```
      SET
      SAVE
       =      L     20.1
      A       M     0.5
      NOT
      JCN     M001
      SET
      O       M     0.5
M001：CLR
      A       L     20.1
      SAVE
      BE
```

**11. 同步错误组织块**

同步错误是与执行用户程序有关的错误，OB121 用于对程序错误的处理，OB122 用于处理模块访问错误。

同步错误 OB 的优先级与检测到出错的块的优先级一致。

同步错误可以用 SFC 36“MASK_FLT”来屏蔽，用错误过滤器中的一位来表示某种同步错误是否被屏蔽。错误过滤器分为程序错误过滤器和访问错误过滤器，分别占一个双字。屏蔽后的错误过滤器可以读出。

可以用 SFC 38“READ_ERR”读出已经发生的被屏蔽的错误。

① 编程错误组织块（OB121）。当有关程序处理的故障事件发生时，CPU 操作系统调用 OB121，OB121 与被中断的块在同一优先级中执行，表 5-15 描述了编程错误 OB121 的临时变量。

**表 5-15　OB121 的变量声明表**

| 变　　量 | 类　　型 | 描　　述 |
|---|---|---|
| OB121_EV_CLASS | BYTE | 事件级别和标识 |
| OB121_SW_FLT | BYTE | 故障代码 |
| OB121_PRIORITY | BYTE | 优先级 = 出现故障的 OB 优先级 |
| OB121_OB_NUMBR | BYTE | OB 号 |
| OB121_BLK_TYPE | BYTE | 出现故障块的类型（在 S7 -300 PLC 时无有效值在这里记录） |
| OB121_RESERVED_1 | BYTE | 备用 |
| OB121_FLT_REG | WORD | 故障源（根据代码）。如：转换故障发生的寄存器；不正确的地址（读/写故障）；不正确的定时器/计数器/块号码；不正确的存储器区 |
| OB121_BLK_NUM | WORD | 引起故障的 MC7 命令的块号码（S7 -300 PLC 无效） |
| OB121_PRG_ADDR | WORD | 引起故障的 MC7 命令的块号码（S7 -300 PLC 无效） |
| OB121_DATE_TIME | DATE_AND_TIME | OB 被调用时的日期和时间 |

OB121 程序在 CPU 执行错误时执行，此错误不包括用户程序的逻辑错误和功能错误等，例如当 CPU 调用一个未下载到 CPU 中的程序块，CPU 会调用 OB121，通过临时变量“OB121_BLK_TYPE”可以得出出现错误的程序块。使用 STEP 7 不能实时监控程序的运行，可以用“变量表（Variable Table）”监控实时数据的变化。

打开事先已经插入的 OB121 编写程序，如图 5-81 所示。

在 SIMATIC 管理器的“块”下，插入 FC1，打开 FC1，并编写程序，如图 5-82 所示。

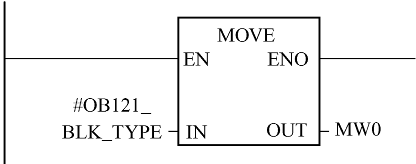



图 5-81　OB121 中编写的程序

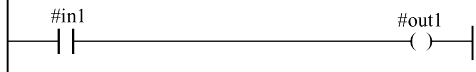



图 5-82　FC1 中编写的程序

打开 OB1 编写程序，如图 5-83 所示。

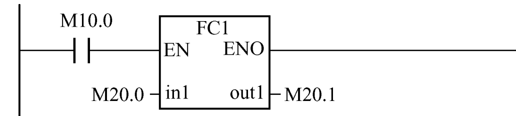



图 5-83　OB1 中编写的程序

先将硬件和 OB1 下载到 CPU 中，此时 CPU 能正常运行。在“块”下插入“变量表（Variable Table）”，然后打开，填入 MW0 和 M10.0，并单击“监控”按钮，程序运行正常。将 M10.0 置为“true”后，CPU 就报错停机，查看 CPU 的诊断缓冲区信息，发现为编程错误，若将 OB121 也下载到 CPU 中，再将 M10.0 置为“true”，CPU 会报错但不停机，MW0 为“W#16#88”，“W#16#88”表示为 OB 程序错误，检查发现 FC1 未下载。下载 FC1 后，在将 M10.0 置为“true”，这时 CPU 不会再报错，程序也不会再调用 OB121。

② I/O 访问错误组织块（OB122）。当对于模块的数据访问出现故障时 CPU 的操作系统调用 OB122，OB122 程序在出现 I/O 访问错误时被调用，例如当 CPU 程序访问一未定义的

I/O 地址，CPU 会出现 I/O 访问错误，CPU 会调用 OB122，如果 OB122 未下载，CPU 会报故障停机。通过临时变量 OB122_SW_FLT 可以读出错误代码，通过 OB122_BLK_TYPE 得出出现错误的程序块，通过 OB122_MEM_AREA 可以读出被访问的地址类型，通过 OB122_MEM_ADDR 可以读出发生错误的存储器地址。表 5-16 描述了 I/O 访问错误 OB122 的临时变量。

**表 5-16　OB122 的变量声明表**

| 变　量 | 类　型 | 描　述 |
|---|---|---|
| OB122_EV_CLASS | BYTE | 事件级别和标识 |
| OB122_SW_FLT | BYTE | 故障代码 |
| OB122_PRIORITY | BYTE | 优先级 = 出现故障的 OB 的优先级 |
| OB122_OB_NUMBR | BYTE | OB 号 |
| OB122_BLK_TYPE | BYTE | 出现故障块的类型（在 S7 -300 PLC 时无有效值在这里记录） |
| OB122_MEM_AREA | BYTE | 存储器区和访问类型：<br>位 7 ~4：访问类型、0：位访问、1：字节访问、2：字访问、3：双字访问；位 3 ~0：存储器区，0：I/O 区、1：输入过程映像、2：输出的过程映像 |
| OB122_MEM_ADDR | WORD | 出现故障的存储器地址 |
| OB122_BLK_NUM | WORD | 引起故障的 MC7 命令的块号码（S7 -300 PLC 无效） |
| OB122_PRG_ADDR | WORD | 引起故障的 MC7 命令的块号码（S7 -300 PLC 无效） |
| OB122_DATE_TIME | DATE_AND_TIME | OB 被调用时的日期和时间 |

以下的指令表程序在组织块 OB122 中，其功能是：当发生 I/O 访问错误时，执行 OB122 中的程序（如下程序），使得 M0.6 得电。如果 M0.6 和 HMI 关联，当发生 I/O 访问错误时，可以发出报警信息，显示在 HMI 上。同时把错误类型（读取还是写入错误）存入 MW22 中，把错误地址存入 MW20 中。

```
        SET
        SAVE
         =      L     20.1
        A       M     0.6
        NOT
        JCN     M001
        SET
         =      M     0.6
M001：  L       #OB122_SW_FLT
        AD      DW#16#FF
        T       MW    22
        L       #OB122_MEM_ADDR
        T       MW    20
        CLR
        A       L     20.1
        SAVE
        BE
```

### 5.5.7　背景组织块

CPU 可以保证设置的最小扫描循环时间，如果它比实际的扫描循环时间长，在循环程

序结束后 CPU 处于空闲的时间内可以执行背景组织块（OB90）。背景 OB 的优先级为 29（最低）。OB90 中的程序一般是对时间要求不严格的程序。

### 5.5.8 启动组织块及其应用

**1. CPU 模块的启动方式**

① 暖启动（Warm Restart）。S7 - 300 CPU（不包括 CPU318）只有暖启动。过程映像数据以及非保持的 M/T/C 将复位。有保持功能的 M/T/C/DB 将保留原数值。模式开关由 STOP 扳到 RUN 位置。

② 热启动（Hot Restart，仅 S7 - 400 PLC 有此方式）。在 RUN 状态时如果突然断电，然后又重新上电，从上次 RUN 模式结束时程序中断之处继续执行，不对计数器等复位。

③ 冷启动（Cold Restart，CPU 417 和 CPU 417H 有此方式）。冷启动时，过程数据区的 I、Q、M、T、C 及 DB 等被复位为零。模式开关扳到 MRES 位置。

**2. 启动组织块（OB100～OB102）应用**

在暖启动、热启动或冷启动时，操作系统分别调用 OB100、OB101 或 OB102。

**【例 5-16】** 编写一段初始化程序，将 CPU 314 - 2DP 的 MB0～MB3 单元清零。

**解：** 一般初始化程序在 CPU 一启动后就运行，CPU 314 - 2DP 只有暖启动方式，所以只能使用 OB100 组织块。MB0～MB3 实际上就是 MD0，其程序如图 5-84 所示。

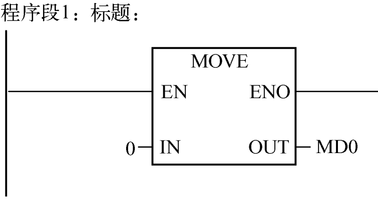



图 5-84 OB100 中编写的程序

**【例 5-17】** 当系统手动暖启动时，Q0.0 亮，当系统自动暖启动时，Q0.0 灭。

**解：** 梯形图如图 5-85 所示。

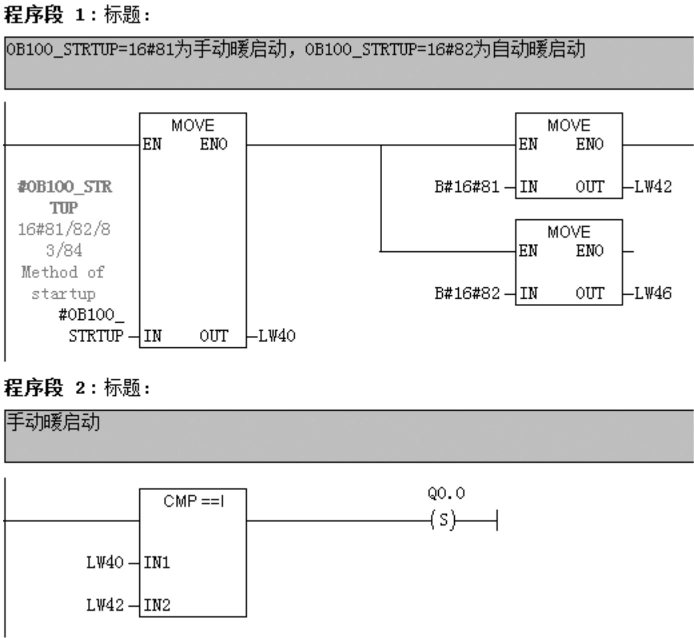



图 5-85 OB100 中编写的梯形图

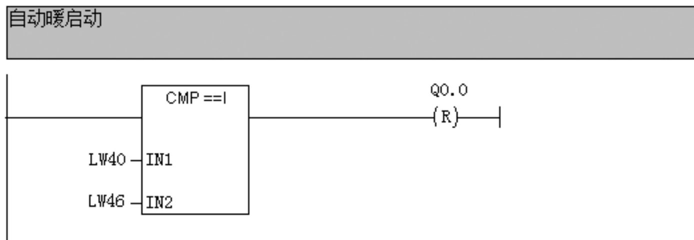



图 5-85　OB100 中编写的梯形图（续）

## 5.6　实例

至此，读者对 S7－300/400 PLC 的软硬件已经有了一定的了解，本节将列举 3 个简单的例子，供读者模仿学习。

**【例 5-18】** 控制系统为 S7－300、ET200、SM323 和 SM332，SM332 输出电压信号，如果电压信号断线时，能通过 STEP 7 的参数表查看到故障信号的通道，要求用 OB82 实现此功能。

**解：**

1）硬件组态过程如下：

首先按照如图 5-86 进行组态，双击"AO2x12Bit"，弹出如图 5-87 所示的界面，在"输出"选项卡中，"勾选""组诊断"和"诊断中断"，最后单击"确定"按钮。

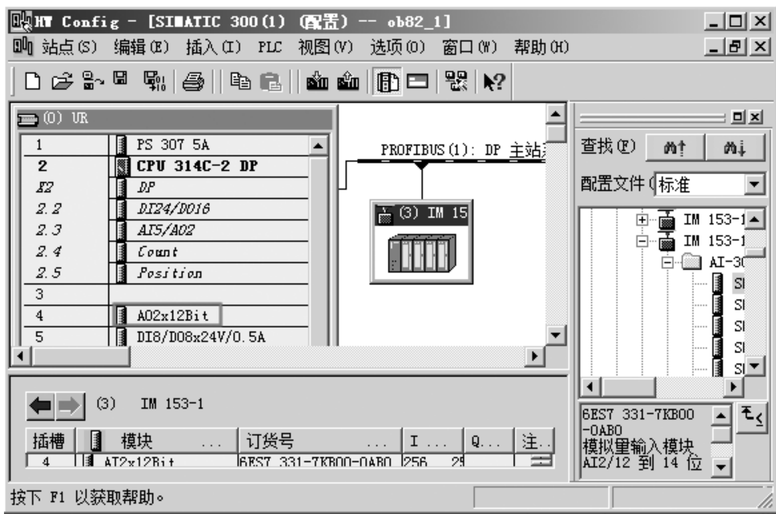



图 5-86　硬件组态（1）

2）创建数据块 DB1，并在 DB1 中创建一个数组 ary[0..20]，数组的容量为 21 个字节，如图 5-88 所示。




图 5-87　硬件组态（2）



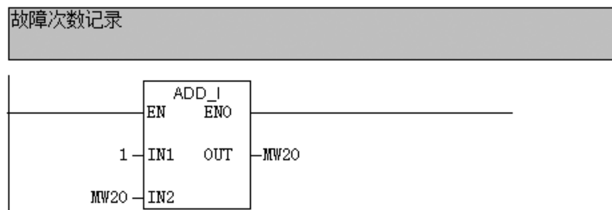

图 5-88　创建的数组

3）创建变量表 VAT_1，并把需要监控的参数输入变量表，如图 5-89 所示，利用这个变量可以监控到故障信息。




图 5-89　创建变量表

4）编写梯形图程序。在 OB82 中编写梯形图程序，如图 5-90 所示。由于 S7-300/400 PLC 有很强的自诊断功能，故障信息自动赋值给 OB82 的参数。程序的作用实际就是将 OB82 的所有参数信息传送到数组 DB1. ary 中，这样故障信息就可以在参数表中显示了。



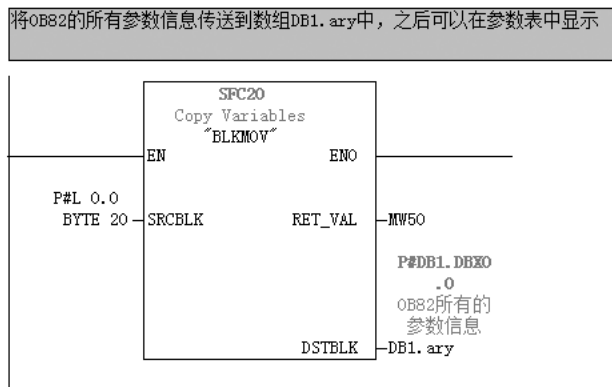




图 5-90 梯形图

5）仿真。将完整的程序下载到仿真器中，将仿真器置于“RUN”状态，在工具栏中单击“Execute”（执行）→“Trigger Error OB”（触发器）→“Diagostic Interrupt”（诊断中断），如图 5-91 所示，打开诊断中断界面。

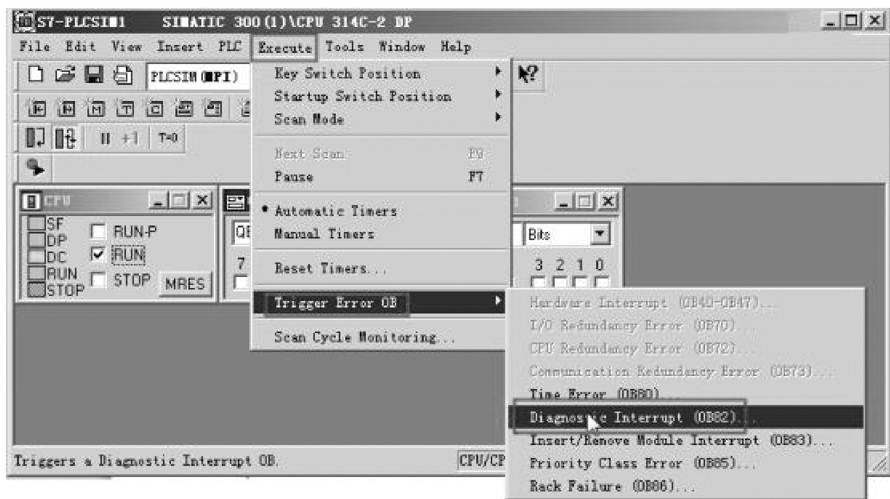



图 5-91 打开“诊断中断”界面

如图 5-92 所示，在“Module address”（模块地址）中输出故障模块的逻辑地址（即模拟量的通道号）“pqw256”，任意选择一个可能的故障，本例选择“Exernall fault”（外部故障），单击“OK”（确定）按钮，模拟外部断线。

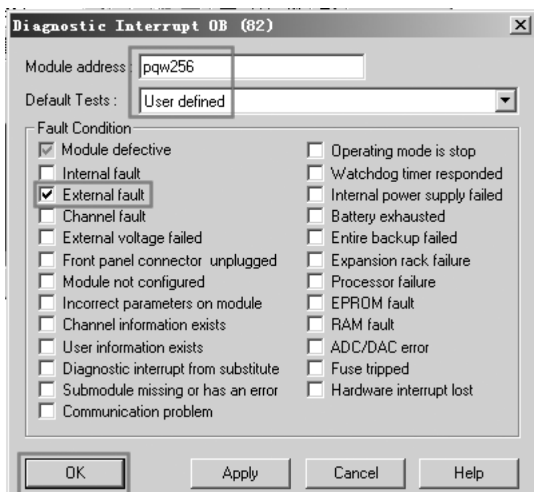



图 5-92　模拟故障

6）监视故障。打开变量表，单击“监视变量”按钮，可以看到多个参数，如图 5-93 所示。参数的含义如下：

变量 - VAT_1

表格(T)　编辑(E)　插入(I)　PLC　变量(A)　视图(V)　选项(O)　窗口(W)　帮助(H)

VAT_1 -- @ob82_1\SIMATIC 300(1)\CPU 314C-2 DP\S7 程序(1)　ON...

| | 地址 | 符号 | 显示格式 | 状态值 | 修改数值 |
|---|---|---|---|---|---|
| 1 | MW 20 | | HEX | W#16#0001 | |
| 2 | DB1.DBD 0 | | HEX | DW#16#39421A52 | |
| 3 | DB1.DBD 4 | | HEX | DW#16#C5550100 | |
| 4 | DB1.DBD 8 | | HEX | DW#16#11050000 | |
| 5 | DB1.DBD 12 | | HEX | DW#16#15020713 | |
| 6 | DB1.DBD 16 | | HEX | DW#16#22427847 | |
| 7 | | | | | |

ob82_1\SIMATIC 300(1)\...\S7 程序(1)　RUN　Abs

图 5-93　监视故障

① MW20 代表故障的累积次数。

② DB1. DBB0 为 16#39，表示事件到来。

③ DB1. DBB5 为 16#55，表示为输出模块。

④ DB1. DBW6 为 16#0100，就是输出模块的地址是 PQW256（256 的十六进制就是 16#0100）。

⑤ DB1. DBW8 为 16#1105（2#0001 0001 0000 0101），外部故障、通道故障及模块发生故障。

**【例 5-19】** 某设备上有一套步进驱动系统，步进驱动器的型号为 SH－2H042Ma，步进电动机的型号为 17HS111，是两相四线直流 24 V 步进电动机，要求：压下按钮 SB1 时，步

进电动机顺时针方向旋转。请画出 I/O 接线图并编写程序。

**解：**

1） 主要软硬件配置。

① 1 套 STEP5.5 SP4。

② 1 台步进电动机，型号为 17HS111。

③ 1 台步进驱动器，型号为 SH－2H042Ma。

④ 1 台 CPU 314C－2DP。

2） 系统的接线图。系统的接线图如图 5-94 所示。

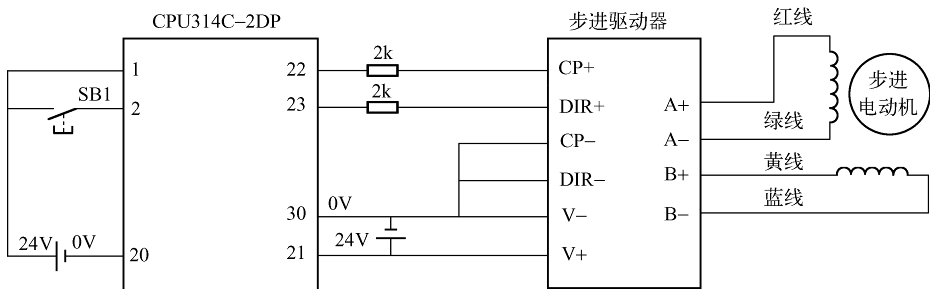



图 5-94　CPU314C－2DP、步进驱动器和步进电动机接线图

3） 硬件组态，过程如下：

从前述的内容知道：使用步进驱动器驱动步进电动机，步进驱动器的脉冲控制信号要求有较高的频率（S7－200 系列 PLC 的 Q0.0 和 Q0.1 就具有此功能），因此能否产生高频脉冲成为能否控制步进驱动器的关键。西门子的 CPU 312C、CPU 313C、CPU 313C－2DP 和 CPU 314C－2DP 等型号的 PLC 集成了用于高速计数和高频脉冲输出的通道，用于高速计数和高频脉冲输出。

CPU314C－2DP 有 4 个高频脉冲输出的特殊通道，这 4 个通道是首字节的最低四位，在一般情况下可以作为普通的数字量输出点来使用。在需要高频脉冲输出时，可通过硬件设置定义这四位的属性，将其作为高频脉冲输出通道来使用。作为普通数字量输出点使用时，其系统默认地址为 Q124.0、Q124.1、Q124.2 和 Q124.3（该地址一般习惯修改成 Q0.0、Q0.1、Q0.2 和 Q0.3），作为高速脉冲输出时，对应的通道分别为 0 通道、1 通道、2 通道和 3 通道（通道号为固定值，用户不能自行修改）。每一通道都可输出最高频率为 2.5 kHz（周期为 0.4 ms）的高频脉冲。

4） 系统的硬件组态。

① 创建一个新项目。首先，起动 STEP5.5，创建一个项目名“步进电动机”的项目，打开其硬件组态界面，如图 5-95 所示。

② 高频脉冲输出属性设置。双击如图 5-96 的“Count”（计数器），弹出计数器属性设置对话框如图 5-97 所示，先选定“Channel”（通道）的 0 号通道，再选定“Operating”（工作模式）中的“Pulse－width modulation”（脉宽调制），此时弹出“Mode change”（模式改变）对话框，单击此对话框上“OK”按钮，最后单击计数器属性设置对话框上的“OK”按钮。

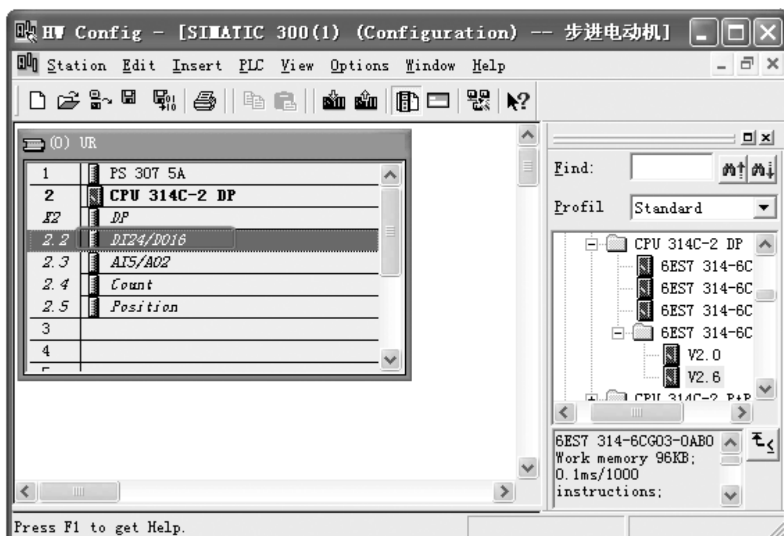



图 5-95　新建项目并打开硬件组态

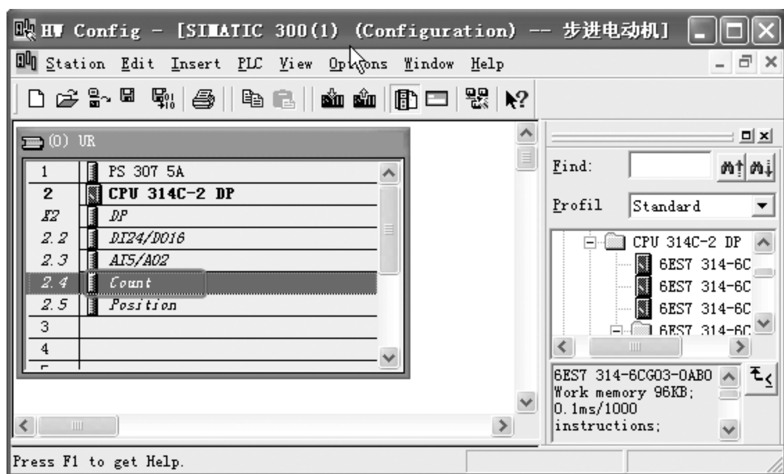



图 5-96　高频脉冲输出属性设置（1）

如图 5-98 所示，先在“Time base”（时基）中选定 0.1ms，再在“On－delay”（接通延时）输入 500，在“period”（周期）中输入 200，接着在“Minimum pulse”（最小脉冲）输入 10，最后单击“OK”按钮。

Operating Parameters（操作参数）中各参数意义如下：

① Output format：Per mile，S7 analog value。输出格式（output－format）有两种选择，每密耳（Per mile）和 S7 模拟量值（S7 analog value）。选择 Per mil，则输出格式取值范围为（0～1000），选择 S7 analog value，则输出格式取值范围为（0～27648，S7 模拟量的最大取值为 27648）。输出格式的取值是在调用系统功能块 SFB49 时设置的，这一取值将会影响输出脉冲的占空比，具体内容将在后面介绍 SFB49 时提到。

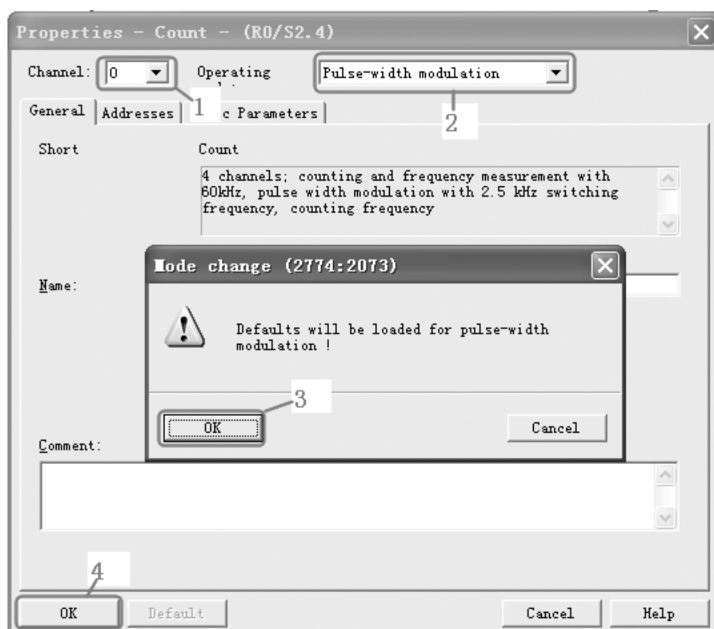



图 5-97　高频脉冲输出属性设置（2）

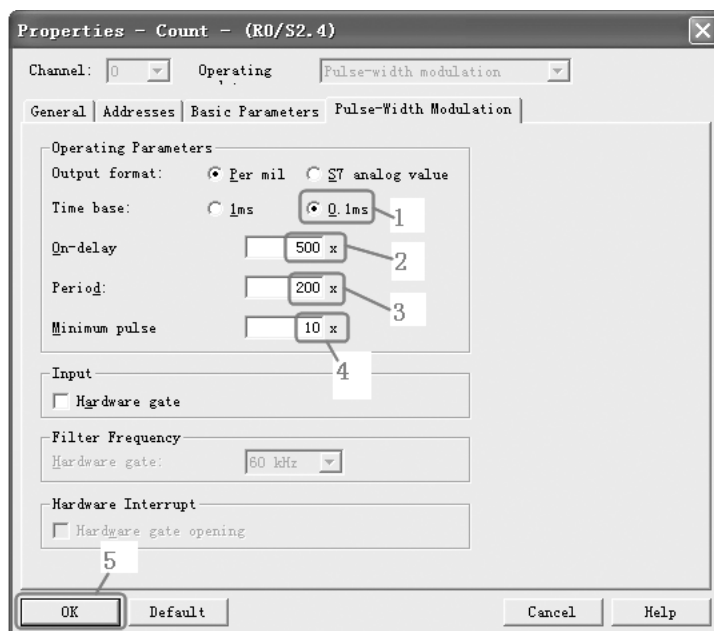



图 5-98　高频脉冲输出属性设置（3）

② Time base：1 ms，0.1 ms。时基（Time base）也有两种选择，用户可根据实际需要选择合适的时基，要产生频率较高的脉冲，可选择较短的时基（0.1 ms）。

③ On－delay：接通延时时间值。接通延时（On－delay）是指当控制条件成立时，对应通道将延时指定时间后输出高频脉冲。指定时间值为设置值 × 时基。取值范围为 0～65535。

④ Period：指定输出脉冲的周期。取值范围为 4 ~ 65535。周期为设置值 × 时基。

⑤ Minimum pulse：指定最小的脉冲宽度。指定输出脉冲的最小脉宽，最小脉宽的取值范围为 2 ~ Period/2（周期值的一半）。注意：在指定了最小脉冲宽度以后，应该保证根据占空比计算出来的高低电平的时间不小于最小脉冲宽度，否则脉冲将不能正常输出。请参看系统功能块 SFB49 使用方法。

⑥ Input ：Hardware gate。通过输入参数选择是否采用硬件门控制，如果选中硬件门前面的方框，则高频脉冲的控制需要硬件门和软件门同时控制，如果不选，则高频脉冲输出单独由软件门控制。具体控制控制方法将在后面的 SFB49 介绍中提到。

⑦ Hardware Interrupt：Hardware gate opening。硬件中断选择，一旦选中硬件门控制以后，此选项将被激活，用户可根据需要选择是否在硬件门起动时刻，调用硬件中断组织块 OB40 中的程序。

5）编写程序。

① 系统功能块 SFB49 简介。编写程序控制步进电动机，需调用系统功能块 SFB49，其参数见表 5-17。

**表 5-17　SFB49 参数表**

| 输入参数 | 数据类型 | 地址 DB | 说　　明 | 取值范围 | 缺省值 |
|---|---|---|---|---|---|
| LADDR | WORD | 0 | 子模块的 I/O 地址，由用户在“HW 配置”中指定。如果 I 和 Q 地址不相等，则必须指定二者中较低的一个 | CPU 专用 | W#16#300 |
| CHANNEL | INT | 2 | 指定的通道号：<br>CPU312C<br>CPU313C<br>CPU314C | 0 ~ 1<br>0 ~ 2<br>0 ~ 3 | 0 |
| SW_EN | BOOL | 4.0 | 软件门：控制脉冲输出 | TRUE/FALSE | FALSE |
| MAN_DO | BOOL | 4.1 | 手动输出控制使能 | TRUE/FALSE | FALSE |
| SET_DO | BOOL | 4.2 | 控制输出 | TRUE/FALSE | FALSE |
| LADDR | WORD | 0 | 子模块的 I/O 地址，由用户在“HW 配置”中指定。如果 I 和 Q 地址不相等，则必须指定二者中较低的一个 | CPU 专用 | W#16#300 |
| CHANNEL | INT | 2 | 指定的通道号<br>CPU312C：<br>CPU313C：<br>CPU314C： | 0 ~ 1<br>0 ~ 2<br>0 ~ 3 | 0 |
| SW_EN | BOOL | 4.0 | 软件门：控制脉冲输出 | TRUE/FALSE | FALSE |
| STS_EN | BOOL | 16.0 | 状态使能端 | TRUE/FALSE | FALSE |
| STS_STRT | BOOL | 16.1 | 硬件门的状态（开始输入） | TRUE/FALSE | FALSE |
| STS_DO | BOOL | 16.2 | 输出状态 | TRUE/FALSE | FALSE |
| JOB_DONE | BOOL | 16.3 | 可以起动新作业 | TRUE/FALSE | TRUE |
| JOB_ERR | BOOL | 16.4 | 故障作业 | TRUE/FALSE | FALSE |
| JOB_STAT | WORD | 18 | 作业错误号 | W#16#0000 至 W#16#FFFF | W#16#0 |

以下对各参数进行介绍。

在 SFB49 的所有输入参数中主要有两部分，一部分是用来控制脉冲输出或作为数字量输出的控制变量；另外一部分则是用来修改脉冲参数，例如脉冲周期、延时时间、最小脉宽等。

参数 LADDR：子模块的地址，数据类型为字。可在硬件组态时进行地址配置。其默认值为 W#16#300，即输入/输出映像区第 768 个字节。若通道集成在 CPU 模块中，则此参数可以不用设置，若通道在某个子功能模块上，则必须保证此参数的地址与模块设置的地址一致。

参数 CHANNEL：通道号，数据类型为整数。此参数指定启用的通道号，CPU313C 具有 3 个通道的高频脉冲输出，通道号分别为 0、1 和 2。如起动 2 号通道，则参数值为：2。若通道号大于 2，则在执行功能块时，将发出出错信息。

参数 SW_EN：软件控制门，数据类型为 BOOL。SFB49 是通过门功能（Gate Function）控制高频脉冲的起动输出的。门功能中包括硬件门（hardware gate）和软件门（software gate）两种，可根据需要设置为单独使用软件门控制或同时使用硬件门和软件门控制。

单独使用软件门控制时，在硬件设置时，不能启用硬件门（hardware gate）控制。此时，高频脉冲输出单独由软件门 SW_EN 端控制，即 SW_EN 端为“1”时，脉冲输出指令开始执行（延时指定时间后输出指定周期和脉宽的高频脉冲），当 SW_EN 端为“0”时，高频脉冲停止输出。

采用硬件门和软件门同时控制时，需要在硬件设置中，启用硬件门控制。当软件门的状态先为“1”，同时在硬件门有一个上升沿时，将起动内部门功能，并输出高频脉冲（延时指定时间输出高频脉冲）。当硬件门的状态先为“1”，而软件门的状态后变为“1”，则门功能不起动，若软件的状态保持“1”，同时在硬件门有一个下降沿发生，也能起动门功能，输出高频脉冲。当软件门的状态变为“0”，无论硬件门的状态如何，将停止脉冲输出。

参数 MAN_DO：手动输出使能端。一旦通道在硬件组态时设置为脉宽调制功能，则该通道不能使用普通的输出线圈指令对其进行写操作控制，要想控制该通道必须调用功能块 SFB49 对其进行控制。如果还想在该通道得到持续的高电平（非脉冲信号），则可以通过 MAN_DO 控制端实现。当 MAN_DO 端为“1”时，指定通道不能输出高频脉冲，只能作为数字量输出点使用。当 MAN_DO 端为“0”时，则指定通道只能作为高频脉冲输出通道使用输出指定频率的脉冲信号。

参数 SET_DO：数字量输出控制端。当 MAN_DO 端的状态为“1”时，可通过 SET_DO 端控制指定通道的状态是为高电平“1”，还是低电平“0”。如果 MAN_DO 端的状态为“0”，则 SET_DO 端的状态不起作用，不会影响通道的状态。

参数 OUTP_VAL：输出值设置。输出值参数数据类型为整数。在硬件设置中我们只指定了脉冲的周期，延时时间以及最小脉宽等，并没有指定脉冲的占空比。参数 OUTP_VAL 就是用来指定脉冲占空比的。注意：在硬件设置时，如果选择输出形式（Output－format）为 Per mil，则 OUTP_VAL 取值范围为 0～1000（基数为 1000），输出脉冲高电平时间长度为

Pulse width(脉宽)＝(OUTP_VAL/1000)×period(周期)。

例如：若 OUTP_VAL 值设置为 200，则一个周期中，200/1000 的时间为高电平，800/1000 的时间为低电平，即占空比为 1∶4。

如果硬件设置时，选择输出形式（output - format）为 S7 模拟量值（S7 analog value），则 OUTP_VAL 取值范围为 0 ~ 27648（基数为 27648），此时输出脉冲的脉宽为

Pulse width(脉宽) = (OUTP_VAL/27648) × period(周期)。

此时如果 OUTP_VAL 值设置为 200，则一个周期中 200/27648 的时间为高电平，27448/27648 的时间为低电平。

参数 JOB_REQ：作业操作信号，通过作业操作可以修改硬件设置时指定的某些参数如延时时间、周期和最小脉宽等。作业初始化控制端是上升沿有效，当 JOB_REQ 端的状态由“0”变为“1”时将进行作业操作功能，具体事件由作业 ID 和作业值决定。

参数 JOB_ID：作业号，作业号决定了具体的作业事件，例如，如果想修改脉冲周期则可指定 JOB_ID 号为 W#16#1，如果想修改延时时间则可指定 JOB_ID 的参数为 W#16#2。如果想读取周期，则指定 JOB_ID 号为 W#16#81。在系统功能 SFB49 的背景数据块中，有一个静态变量：JOB_OVAL，变量类型为双整数，SFB49 进行读作业操作时，将把读取的值放在这一区域，用户可访问这一区域得到高频脉冲相关参数的值。

参数 JOB_ERR：故障作业，当 JOB_ERR 的状态为“0”时，表示作业执行正常，当 JOB_ERR 的状态为“1”时，表示有故障发生，具体故障原因可查看故障代码（由 JOB_STAT 端读取）

参数 JOB_STAT：作业错误代码。如果发生作业错误，则 JOB_ERR = TRUE。JOB_STAT 中将给出精确的错误原因。

② 打开组织块 OB1，如图 5-99 所示，在 Library/Standard Library/System Function Blocks 菜单下双击“SFB49”，并调用这个系统块。

③ 在 OB1 中输入如图 5-100 所示程序，并保存程序。




图 5-99　打开组织块

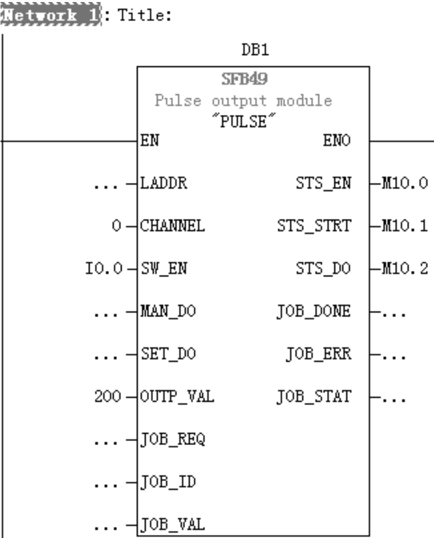



图 5-100　程序

**【例 5-20】**某控制系统，控制器是 CPU 314C - 2DP，压力传感器测量油压力，油压力的范围是 0 ~ 10 MPa，当油压力高于 8 MPa 时报警，请设计此系统。

**解**：CPU 314C - 2DP 集成了有模拟量输入/输出和数字量输入/输出，其接线如图 5-101

所示，模拟量输入的端子2和4分别与传感器的电压信号+和电压信号-相连，传感器的电源在图中未表示。

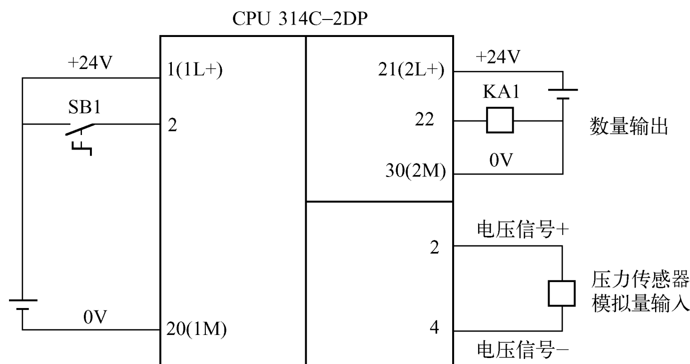



图5-101 接线图

数值转换（FC105）SCALE功能接受一个整型值（IN），并将其转换为以工程单位表示的介于下限和上限（LO_LIM和HI_LIM）之间的实型值。将结果写入OUT。SCALE功能使用以下等式：

$$OUT=\frac{IN-K1}{K2-K1}(HI_LIM-LO_LIM)+LO_LIM$$

常数K1和K2根据输入值是BIPOLAR（双极性）还是UNIPOLAR（单极性）设置。为BIPOLAR时，假定输入整型值介于-27648~27648之间，因此K1=-27648.0，K2=+27648.0；为UNIPOLAR时，假定输入整型值介于0~27648之间，因此K1=0.0，K2=+27648.0。

本例是单极性，故K1=0.0，K2=+27648.0。上极限HI_LIM为10和下极限LO_ LIM为0。模拟量采集并转换到PIW256中，这个数值的范围是0~27648，经过数值转化后，MW14（OUT）中的数值为0~10，是油压力。这个题目也可以不用转换函数FC105，而直接用数学函数，但要复杂得多。

梯形图如图5-102和图5-103所示。

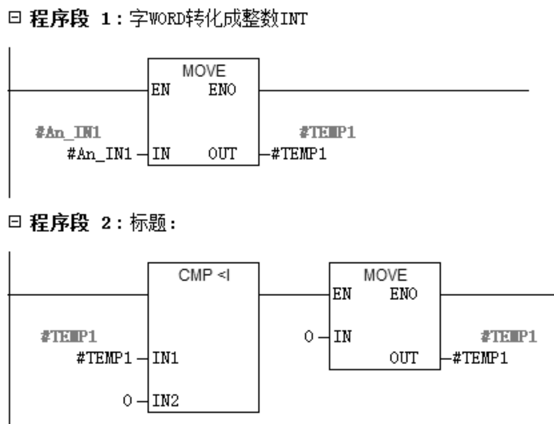
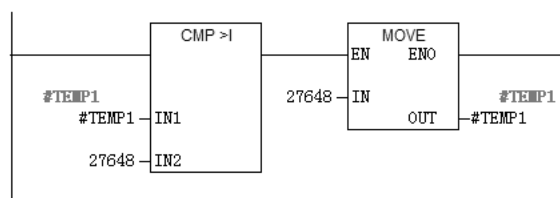



图5-102 FC1中的程序

⊟ **程序段 3**：标题：




⊟ **程序段 4**：标题：

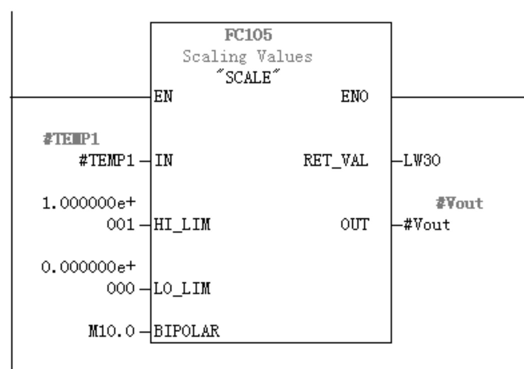



图 5-102　FC1 中的程序（续）

⊟ **程序段 1**：标题：




⊟ **程序段 2**：标题：

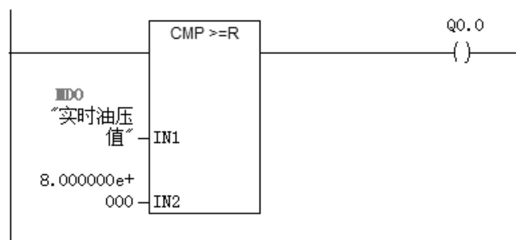



图 5-103　OB35 中的程序

# 第 6 章　S7－300/400 PLC 的编程方法与调试

本章介绍功能图的画法、梯形图的禁忌以及如何根据功能图用基本指令、功能指令和复位置位指令编写顺序控制梯形图。另一个重要的内容是程序的调试方法。

## 6.1　功能图

### 6.1.1　功能图的画法

功能图（SFC）是描述控制系统的控制过程、功能和特征的一种图解表示方法。它具有简单、直观等特点，不涉及控制功能的具体技术，是一种通用的语言，也是 IEC（国际电工委员会）首选的编程语言，近年来在 PLC 的编程中已经得到了普及与推广。在 IEC 60848 中称顺序功能图，在我国国家标准 GB 6988－2008 中称功能表图。西门子称为图形编程语言 S7－Graph 和 S7－HiGraph。

顺序功能图是设计 PLC 顺序控制程序的一种工具，适合于系统规模较大，程序关系较复杂的场合，特别适合于对顺序操作的控制。在编写复杂的顺序控制程序时，采用 S7－Graph 和 S7－HiGraph 比梯形图更加直观。

功能图的基本思想是：设计者按照生产要求，将被控设备的一个工作周期划分成若干个工作阶段（简称“步”），并明确表示每一步要执行的输出，“步”与“步”之间通过制定的条件进行转换，在程序中，只要通过正确连接进行“步”与“步”之间的转换，就可以完成被控设备的全部动作。

PLC 执行功能图程序的基本过程是：根据转换条件选择工作“步”，进行“步”的逻辑处理。组成功能图程序的基本要素是步、转换条件和有向连线，如图 6-1 所示。

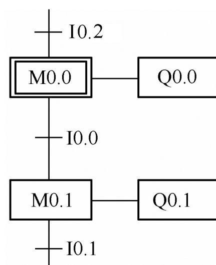



图 6-1　功能图

**1. 步**

一个顺序控制过程可分为若干个阶段，也称为“步”或状态。系统初始状态对应的“步”称为初始步，初始步一般用双线框表示。在每一步中施控系统要发出某些“命令”，而被控系统要完成某些“动作”，“命令”和“动作”都称为动作。当系统处于某一工作阶段时，则该“步”处于激活状态，称为“活动步”。

**2. 转换条件**

使系统由当前步进入下一步的信号称为转换条件。顺序控制设计法用转换条件控制代表各步的编程元件，让它们的状态按一定的顺序变化，然后用代表各步的编程元件去控制输出。不同状态的“转换条件”可以不同，也可以相同。当“转换条件”各不相同时，在功能图程序中每次只能选择其中一种工作状态（称为“选择分支”），当“转换条件”都相同时，在功能图程序中每次可以选择多个工作状态（称为“选择并行分支”）。只有满足条件

状态，才能进行逻辑处理与输出。因此，“转换条件”是功能图程序选择工作状态（步）的“开关”。

**3. 有向连线**

步与步之间的连接线称为“有向连线”，“有向连线”决定了状态的转换方向与转换途径。在有向连线上有短线，表示转换条件。当条件满足时，转换得以实现，即上一步的动作结束而下一步的动作开始，因而不会出现动作重叠。步与步之间必须要有转换条件。

图 6-1 中的双框为初始步，M0.0 和 M0.1 是步名，I0.0、I0.1 为转换条件，Q0.0、Q0.1 为动作。当 M0.0 有效时，输出指令驱动 Q0.0。步与步之间的连线称为有向连线，它的箭头省略未画。

**4. 功能图的结构分类**

根据步与步之间的进展情况，功能图分为以下 5 种结构。

（1）单一顺序

单一顺序动作是一个接一个地完成，完成每步只连接一个转移，每个转移只连接一个步，如图 6-3 和图 6-4 所示的功能图和梯形图是一一对应的。以下用“起保停电路”来讲解功能图和梯形图的对应关系。

为了便于将顺序功能图转换为梯形图，采用代表各步的编程元件的地址（比如 M0.2）作为“步”的代号，并用编程元件的地址来标注转换条件和各步的动作和命令，当某步对应的编程元件置 1，代表该步处于活动状态。

①“起保停”电路对应的布尔代数式。标准的“起保停”梯形图如图 6-2 所示，图中 I0.0 为 M0.2 的起动条件，当 I0.0 置 1，M0.2 得电；I0.1 为 M0.2 的停止条件，当 I0.1 置 1，M0.2 断电；M0.2 的辅助触头为 M0.2 的保持条件。该梯形图对应的布尔代数式为

$$M0.2 = (I0.0 + M0.2) \cdot \overline{I0.1}$$

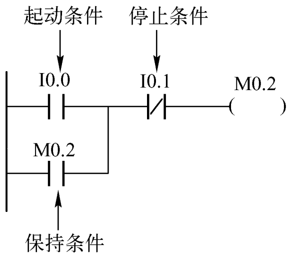



图 6-2　标准的“起保停”梯形图

② 顺序控制梯形图储存位对应的布尔代数式。如图 6-3a 所示的功能图，M0.1 转换为活动步的条件是 M0.1 步的前一步是活动步，相应的转换条件（I0.0）得到满足，即 M0.1 的起动条件为 M0.0 * I0.0。当 M0.2 转换为活动步后，M0.1 转换为不活动步，因此，M0.2 可以看成 M0.1 的停止条件。由于大部分转换条件都是瞬时信号，即信号持续的时间比他激活的后续步的时间短，因此应当使用有记忆功能的电路控制代表步的储存位。在这情况下，起动条件、停止条件和保持条件全部具备，就可以采用“起保停”方法设计顺序功能图的布尔代数式和梯形图。顺序控制功能图中储存位对应的布尔代数式如图 6-3b 所示，参照图 6-2 所示的标准“起保停”梯形图，就可以轻松地将图 6-3 所示的顺序功能图转换为如图 6-4 所示的梯形图。

（2）选择顺序

选择顺序是指某一步后有若干个单一顺序等待选择，称为分支，一般只允许选择进入一个顺序，转换条件只能标在水平线之下。选择顺序的结束称为合并，用一条水平线表示，水平线以下不允许有转换条件，如图 6-5 所示。

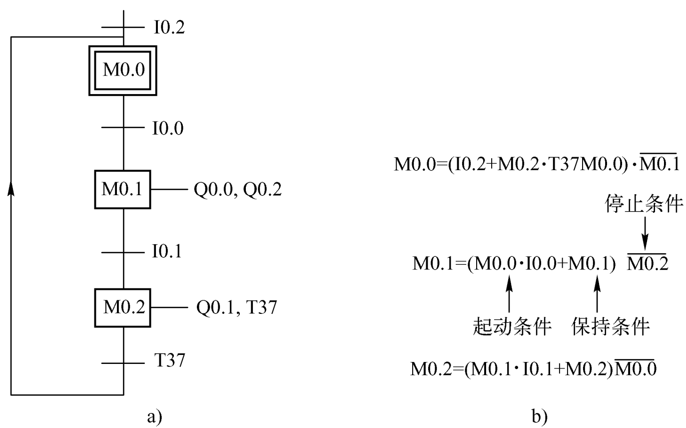
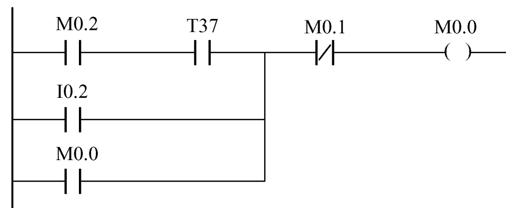


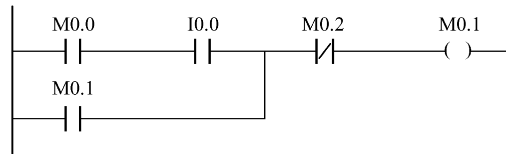

图 6-3 顺序功能图和对应的布尔代数式

a）功能图 b）布尔代数式

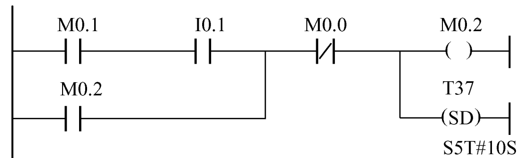
程序段1：标题：



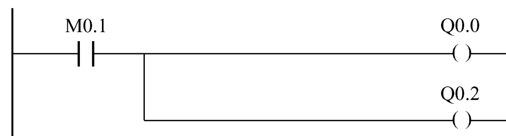

程序段2：标题：




程序段3：标题：




程序段4：标题：




程序段5：标题：

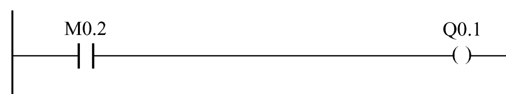



图 6-4 梯形图

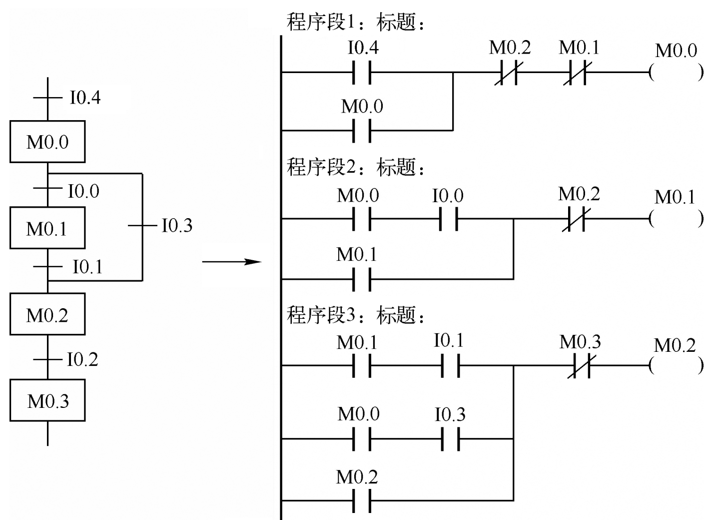



图 6-5　选择顺序

（3）并行顺序

并行顺序是指在某一转换条件下同时起动若干个顺序，也就是说转换条件实现导致几个分支同时激活。并行顺序的开始和结束都用双水平线表示，如图 6-6 所示。

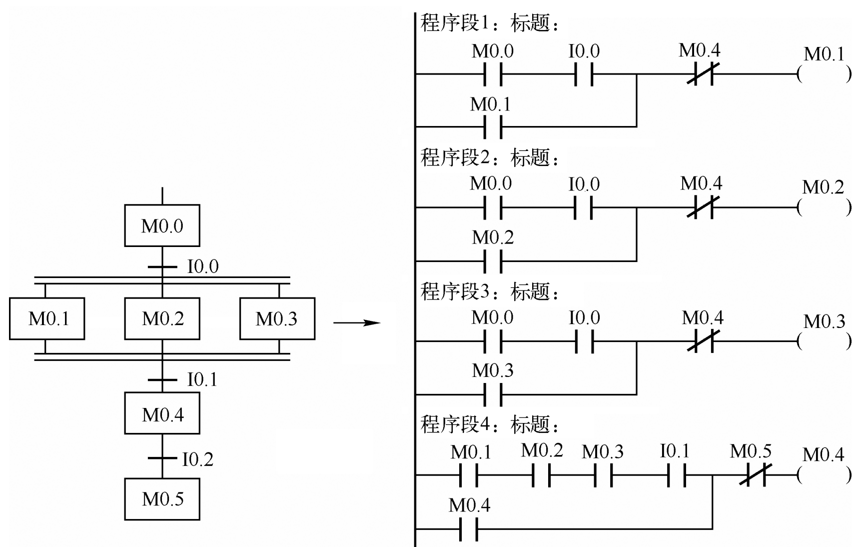



图 6-6　并行顺序

（4）选择序列和并行序列的综合

如图 6-7 所示，步 M0.0 之后有一个选择序列的分支，设 M0.0 为活动步，当它的后续步 M0.1 或 M0.2 变为活动步时，M0.0 变为不活动步，即 M0.0 为 0 状态，所以应将 M0.1 和 M0.2 的常闭触头与 M0.0 的线圈串联。

步 M0.2 之前有一个选择序列合并，当步 M0.1 为活动步（即 M0.1 为 1 状态），并且转换条件 I0.1 满足，或者步 M0.0 为活动步，并且转换条件 I0.2 满足，步 M0.2 变为活动步，所以该步的存储器 M0.2 的“起保停”电路的起动条件为 M0.1 · I0.1 + M0.0 · I0.2，对应

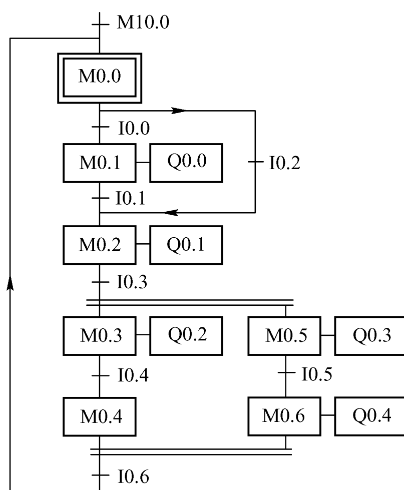
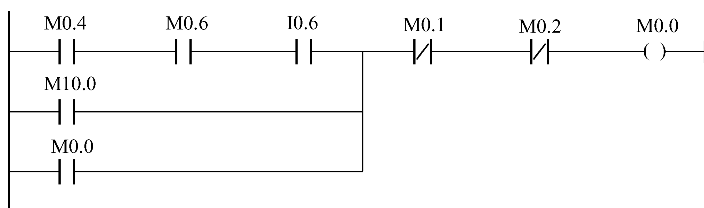


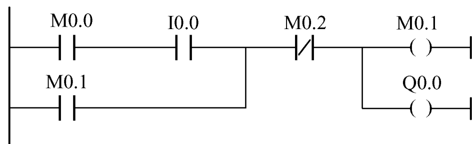

图 6-7　选择序列和并行序列功能图

的起动电路由两条并联支路组成。

步 M0.2 之后有一个并行序列分支，当步 M0.2 是活动步并且转换条件 I0.3 满足时，步 M0.3 和步 M0.5 同时变成活动步，这时用 M0.2 和 I0.3 常开触头组成的串联电路，分别作为 M0.3 和 M0.5 的起动电路来实现，与此同时，步 M0.2 变为不活动步。

步 M0.0 之前有一个并行序列的合并，该转换实现的条件是所有的前级步（即 M0.4 和 M0.6）都是活动步和转换条件 I0.6 满足。由此可知，应将 M0.4、M0.6 和 I0.6 的常开触头串联，作为控制 M0.0 的“起保停”电路的起动电路。图 6-7 所示的功能图对应的梯形图如图 6-8 所示。



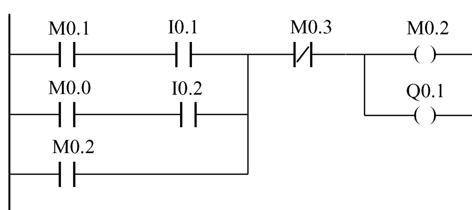
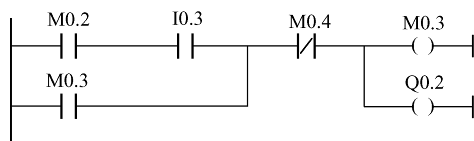
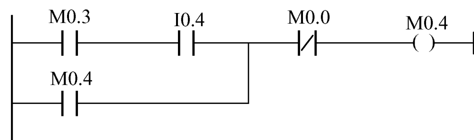




图 6-8　梯形图

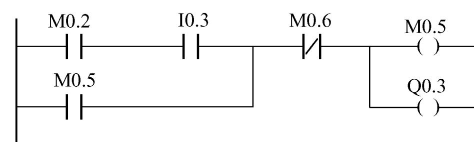
程序段4：标题：




程序段5：标题：




程序段6：标题：




程序段7：标题：

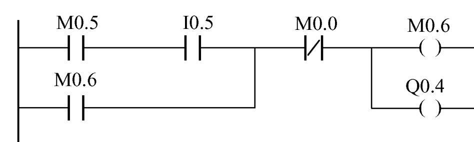



图 6-8　梯形图（续）

**5. 功能图设计的注意点**

1）状态之间要有转换条件。如图 6-9 所示，状态之间缺少“转换条件”是不正确的，应改成如图 6-10 所示的功能图。必要时转换条件可以简化，如将图 6-11 简化成图 6-12。

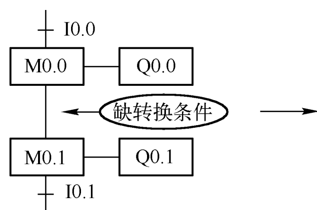



图 6-9　错误的功能图

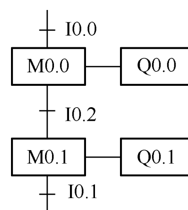



图 6-10　正确的功能图

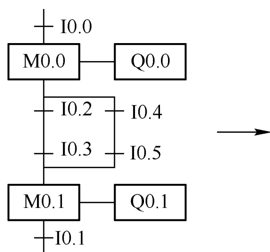



图 6-11　简化前的功能图

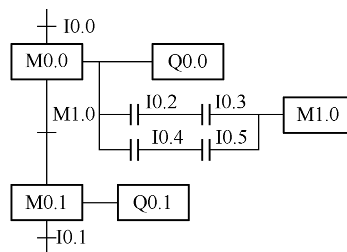



图 6-12　简化后的功能图

2）转换条件之间不能有分支。例如，图 6-13 应该改成如图 6-14 所示的合并后的功能图，合并转换条件。

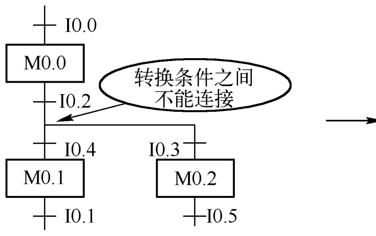



图 6-13　错误的功能图

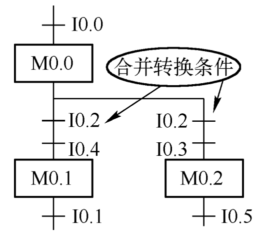



图 6-14　合并后的功能图

3）顺序功能图中的初始步对应于系统等待起动的初始状态，初始步是必不可少的。

4）顺序功能图中一般应有由“步”和有向连线组成的闭环。

## 6.1.2　梯形图编程的原则

尽管梯形图与继电器电路图在结构形式、元件符号及逻辑控制功能等方面相类似，但它们又有许多不同之处，梯形图有自己的编程规则。

1）每一逻辑行总是起于左母线，最后终止于线圈或右母线（右母线可以不画出），如图 6-15 所示。

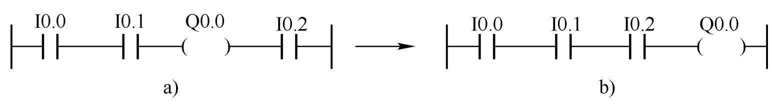



图 6-15　梯形图
a）错误　b）正确

2）无论选用哪种机型的 PLC，所用元件的编号必须在该机型的有效范围内。例如 S7 - 300 PLC 中没有 M99000. 0。

3）梯形图中的触点可以任意串联或并联，但线圈只能并联而不能串联。

4）触点的使用次数不受限制。例如，辅助继电器 M0. 0 可以在梯形图中出现无限制的次数，而实物继电器的触点一般少于 8 对，只能用有限次。

5）在梯形图中同一线圈只能出现一次。如果在程序中，同一线圈使用了两次或多次，称为“双线圈输出”。对于“双线圈输出”，有些 PLC 将其视为语法错误，绝对不允许（如三菱 FX 系列 PLC）；有些 PLC 则将前面的输出视为无效，只有最后一次输出有效（如西门子 PLC）；而有些 PLC 在含有跳转指令或步进指令的梯形图中允许双线圈输出。

6）西门子 PLC 的梯形图中不能出现 I 线圈。

7）对于不可编程的梯形图必须经过等效变换，变成可编程梯形图，如图 6-16 所示。

8）在有几个串联电路相并联时，应将串联触点多的回路放在上方，归纳为“多上少下”的原则，如图 6-17 所示。在有几个并联电路相串联时，应将并联触点多的回路放在左方，归纳为“多左少右”原则，如图 6-18 所示。因为这样所编制的程序简洁明了，语句较少。但要注意图 6-17a 和图 6-18a 的梯形图逻辑上是正确的。

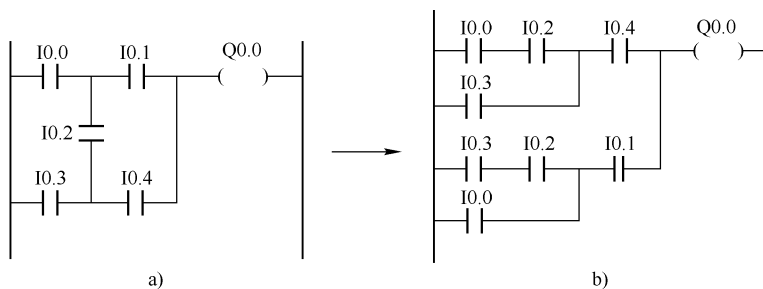



图 6-16　梯形图

a）错误　b）正确

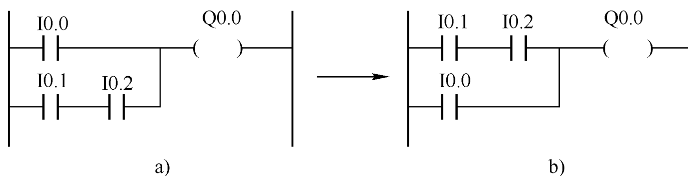



图 6-17　梯形图

a）不合理　b）合理

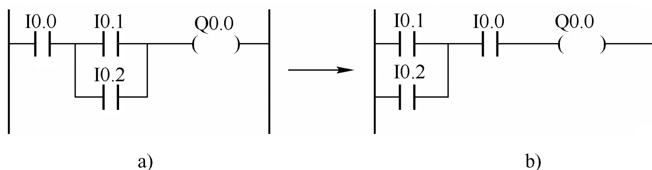



图 6-18　梯形图

a）不合理　b）合理

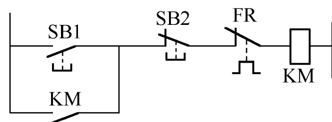



图 6-19　电动机起/停控制图

9）PLC 的输入端所连的电器元件通常使用常开触点，即使与 PLC 对应的继电器 - 接触器系统原来使用的是常闭触点，改为 PLC 控制时也应转换为常开触点。如图 6-19 所示为继电器 - 接触器系统控制的电动机的起/停控制，如图 6-20 所示为电动机的起/停控制的梯形图，如图 6-21 所示为电动机起/停控制的接线图。可以看出：继电器 - 接触器系统原来使用常闭触点 SB1 和 FR，改用 PLC 控制时，则在 PLC 的输入端变成了常开触点。

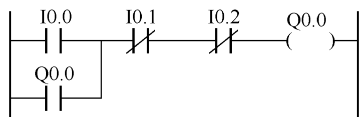



图 6-20　电动机起/停控制的梯形图

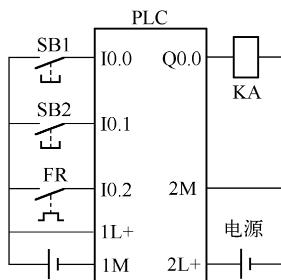



图 6-21　电动机的起/停控制的接线图

【关键点】图 6-20 的梯形图中 I0.1 和 I0.2 用常闭触点，否则控制逻辑不正确。若要 PLC 的输入端的按钮为常闭触点接入也可以，但梯形图中 I0.1 和 I0.2 要用常开触点，对于急停按钮必须使用常闭触头，若一定要使用常开触头，从逻辑上讲是可行的，但在某些情况下，有可能急停按钮不起作用而造成事故，这是读者要特别注意的。另外，一般不推荐将热继电器的常开触点接在 PLC 的输入端，因为这样做占用了宝贵的输入点，最好将热继电器的常闭触点接在 PLC 的输出端，与 KA 的线圈串联。

## 6.2 逻辑控制的梯形图编程方法

相同的硬件系统，由不同的人设计，可能设计出不同的程序，有的人设计的程序简洁、而且可靠，而有的人设计的程序虽然能完成任务，但较复杂，PLC 程序设计是有规律可循的，下面将介绍两种方法：经验设计法和功能图设计法。

### 6.2.1 经验设计法

经验设计法就是在一些典型的梯形图的基础上，根据具体的对象对控制系统的具体要求，对原有的梯形图进行修改和完善。这种方法适合有一定工作经验的人，这些人有现成的资料，特别在产品更新换代时，使用这种方法比较节省时间。下面举例说明这种方法的思路。

**【例 6-1】**图 6-22 所示为小车运输系统的示意图和 I/O 接线图，SQ1、SQ2、SQ3 和 SQ4 是限位开关，小车先左行，在 SQ1 处装料，10 s 后右行，到 SQ2 后停止卸料 10 s 后左行，碰到 SQ1 后停下装料，就这样不停地循环工作。限位开关 SQ3 和 SQ4 的作用是当 SQ2 或者 SQ1 失效时，SQ3 和 SQ4 起保护作用，SB1 和 SB2 是起动按钮，SB3 是停止按钮。

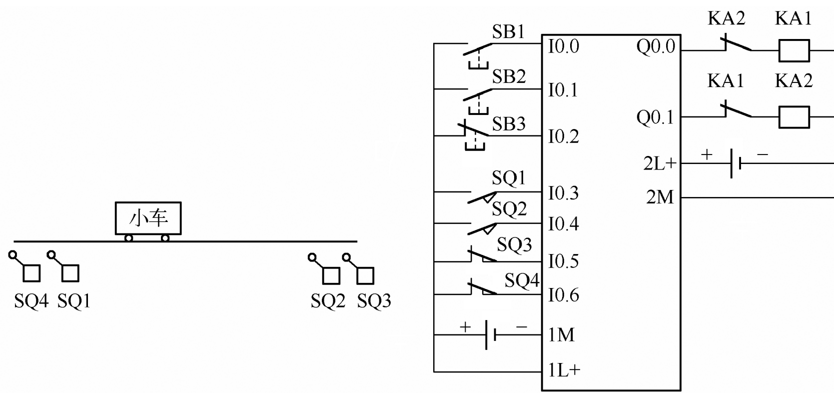
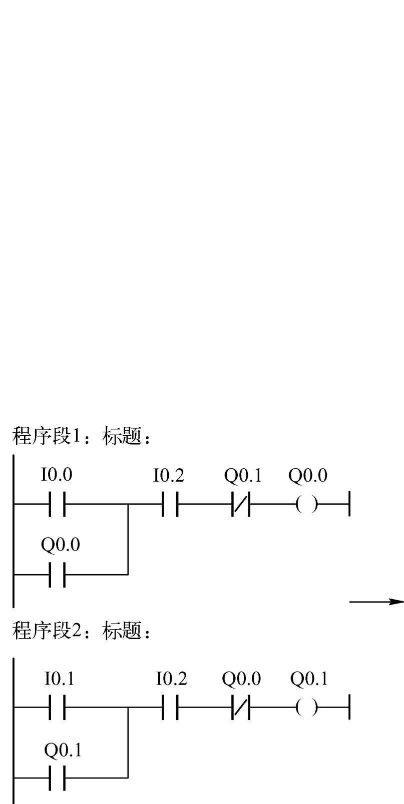



图 6-22　小车运输系统的示意图和 I/O 接线图

**解：**小车左行和右行是不能同时进行的，因此有联锁关系，与电动机的正、反转的梯形图类似，因此先画出电动机正、反转控制的梯形图，如图 6-23 所示，再在这个梯形图的基础上进行修改，增加 4 个限位开关的输入，增加两个定时器，就变成了图 6-24 所示的梯形图。



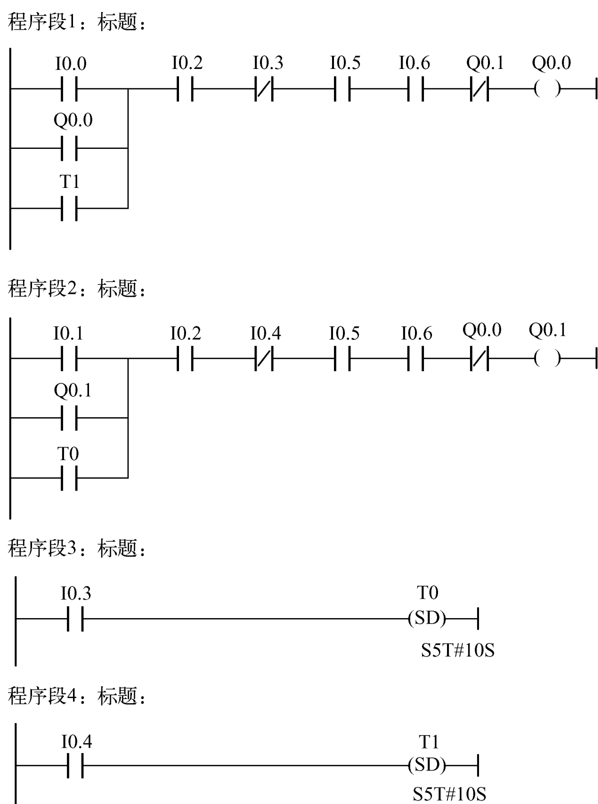




图 6-23　电动机正、反转控制的梯形图

图 6-24　小车运输系统的梯形图

## 6.2.2　功能图设计法

功能图设计法也称为“起保停”设计法。对于比较复杂的逻辑控制，用经验设计法就不合适，适合用功能图设计法。功能图设计法无疑是应用较为广泛的设计方法。功能图就是顺序功能图，功能图设计法就是先根据系统的控制要求设计出功能图，再根据功能图设计梯形图，梯形图可以是基本指令梯形图，也可以是顺控指令梯形图和功能指令梯形图。因此，设计功能图是整个设计过程的关键，也是难点。

**1. “起保停”设计方法的基本步骤**

1）绘制出顺序功能图。要使用“起保停”设计方法设计梯形图时，先要根据控制要求设计出顺序功能图，其中顺序功能图的设计在前面章节中已经详细讲解，在此不再重复。

2）写出储存器位的布尔代数式。对应于顺序功能图中的每一个储存器位都可以写出如图 6-25 所示的布尔代数式。图中等号左边的 $M_i$为第 i 个储存器位的状态，等号左边的 $M_i$为第 i 个储存器位的常开触头，$X_i$为第 i 个工步所对应的转换信号，$M_{i-1}$为第 i－1 个储存器位的常开触头，$M_{i+1}$为第 i＋1 个储存器位的常闭触头。

3）写出执行元件的逻辑函数式。执行元件为顺序功能图中的储存器位所对应的动作。一个“步”通常对应一个动作，输出和对应“步”的储存器位的线圈并联或者在输出线圈

前串接一个对应“步”的储存器位的常开触头。当功能图中有多个“步”对应同一动作时，其输出可用这几个“步”对应的储存器位的“或”来表示，如图 6-26 所示。




图 6-25　存储器位的布尔代数式　　图 6-26　多个“步”对应同一动作时的梯形图

4）设计梯形图。在完成前 3 个步骤的基础上，可以设计出梯形图。

**2. 利用基本指令编写梯形图指令**

用基本指令编写梯形图指令是最常规的设计方法，不必掌握过多的指令。采用这种方法编写程序的过程是：先根据控制要求设计正确的功能图，再根据功能图写出正确的布尔表达式，最后根据布尔表达式编写基本指令梯形图。以下通过例子讲解利用基本指令编写梯形图指令的方法。

**【例 6-2】**步进电动机是一种将电脉冲信号转换为电动机旋转角度的执行机构。当步进驱动器接收到一个脉冲，就驱动步进电动机按照设定的方向旋转一个固定的角度（称为步距角）。步进电动机是按照固定的角度一步一步转动的，因此可以通过脉冲数量控制步进电机的运行角度，并通过相应的装置，控制运动的过程。对于四相八拍的步进电动机。其控制要求如下：

1）按下起动按钮，定子磁极 A 通电，1 s 后 A、B 同时通电；再过 1 s，B 通电，同时 A 失电；再过 1 s，B、C 同时通电……，以此类推，其通电过程如图 6-27 所示。

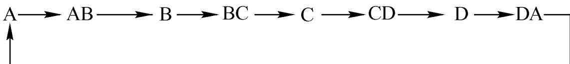



图 6-27　通电过程图

2）有两种工作模式。工作模式 1 时，按下“停止”按钮，完成一个工作循环后，停止工作；工作模式 2 时，具有锁相功能，当按下“停止”按钮后，停止在通电的绕组上，下次按下“起动”按钮时，从上次停止的线圈开始通断电工作。

3）无论何种工作模式，只要按下“急停”按钮，系统所有线圈立即断电。

**解**：接线图如图 6-28 所示，根据题意很容易设计出功能图，如图 6-29 所示。如图 6－30 为初始化程序，根据功能图编写梯形图程序如图 6-31 所示。

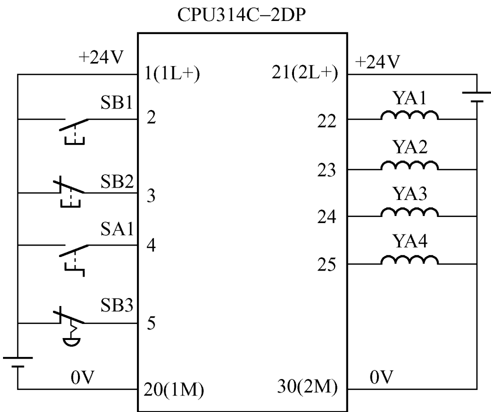



图 6-28　接线图

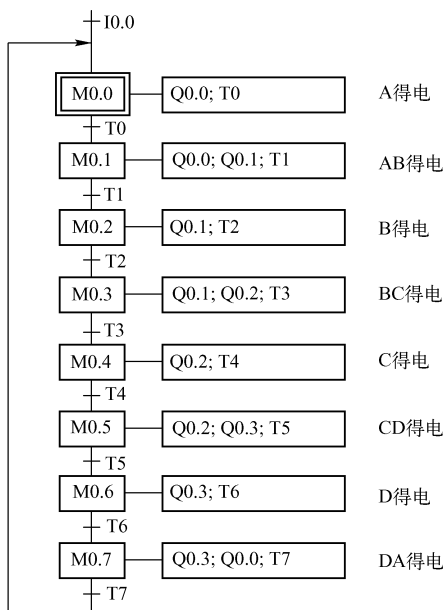



图 6-29　功能图

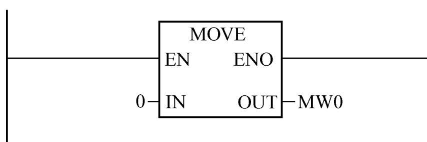
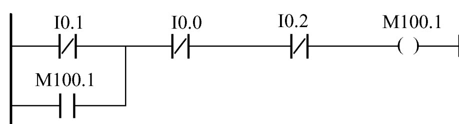
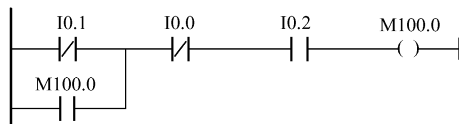
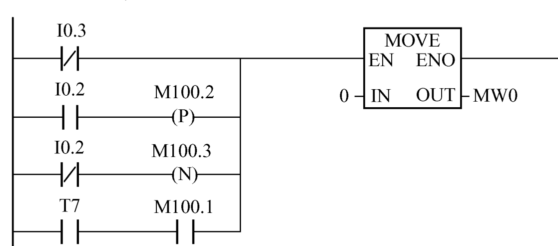



图 6-30　OB100 中的程序



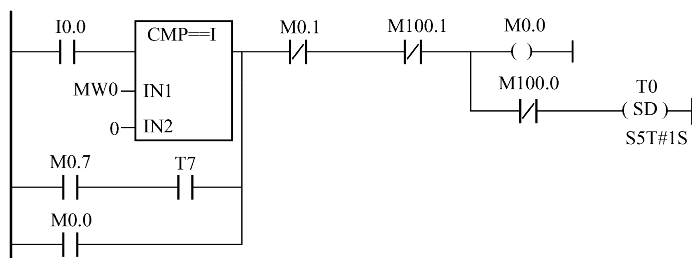
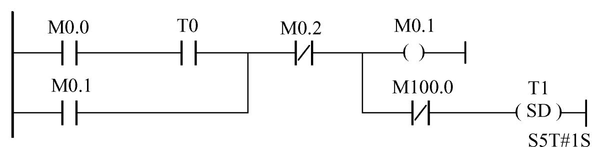
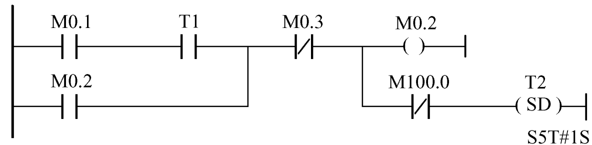




图 6-31　OB1 中的程序

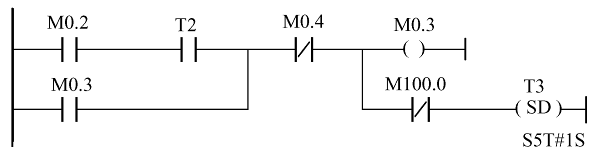
程序段5：标题



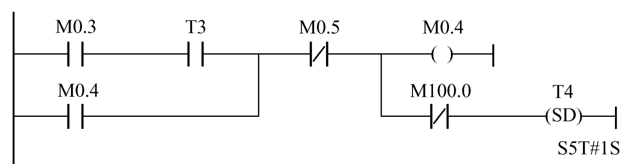

程序段6：标题



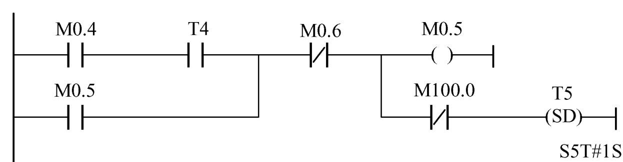

程序段7：标题



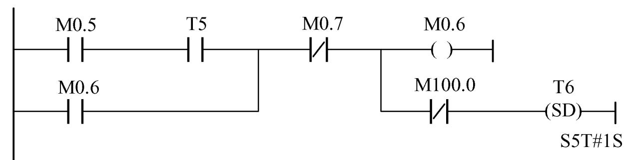

程序段8：标题：



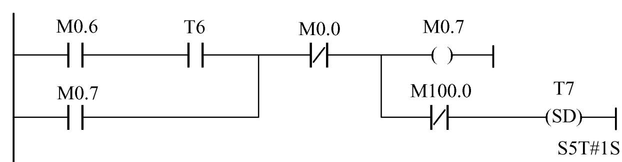

程序段9：标题：




程序段10：标题：




程序段11：标题：




程序段12：标题：




图 6-31　OB1 中的程序（续）

程序段13：标题：




程序段14：标题：




程序段15：标题：




图 6-31　OB1 中的程序（续）

**3. 利用功能指令编写逻辑控制程序**

西门子的功能指令有许多特殊功能，其中移位指令和循环指令非常适合用于顺序控制，用这些指令编写程序简洁而且可读性强。以下通过例子来讲解如何利用功能指令编写逻辑控制程序。

程序段1：标题：

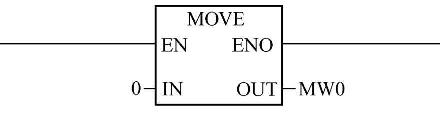
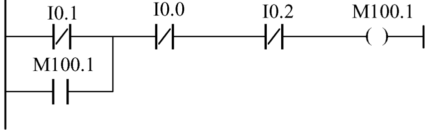



图 6-32　OB100 中的程序

**【例 6-3】**用功能指令编写例 6-2 的程序。

**解：**梯形图如图 6-32 和图 6-33 所示。

程序段1：模式1




程序段2：模式2

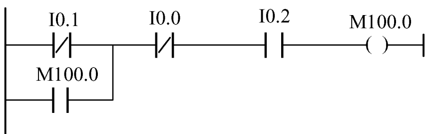
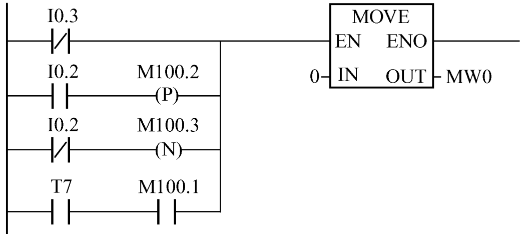
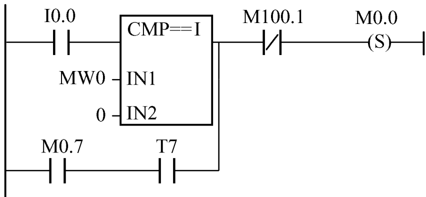



图 6-33　OB1 中的程序

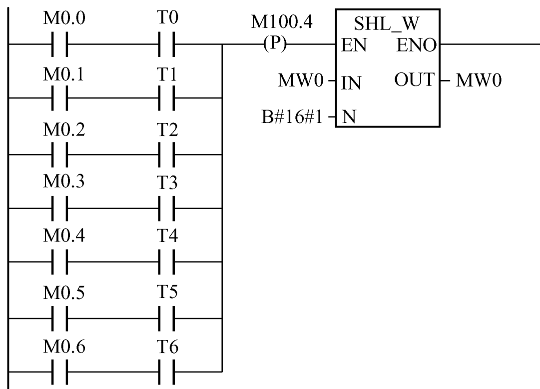
程序段3：急停和模式转换



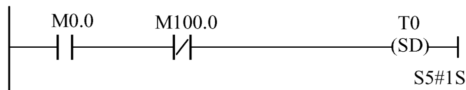

程序段4：标题



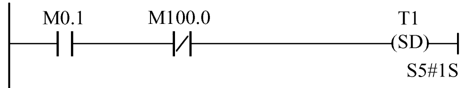

程序段5：标题



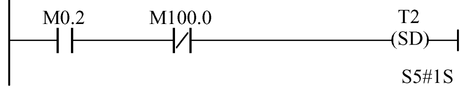

程序段6：标题：




程序段7：标题：




程序段8：标题：




程序段9：标题：

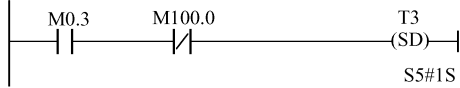
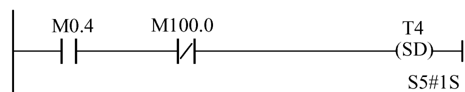
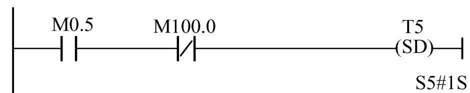



图 6-33　OB1 中的程序（续）

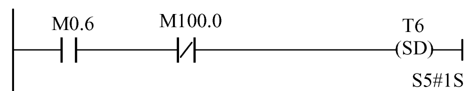
程序段10：标题：



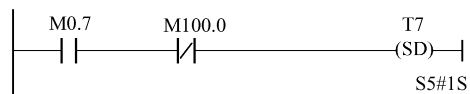

程序段11：标题：




程序段12：标题：




程序段13：标题：




程序段14：标题：




程序段15：标题：




程序段16：标题：




图 6-33　OB1 中的程序（续）




图 6-33　OB1 中的程序（续）

### 4. 利用复位和置位指令编写逻辑控制程序

复位和置位指令是常用指令，用复位和置位指令编写程序简洁而且可读性强。以下通过例子讲解如何利用复位和置位指令编写逻辑控制程序。

**【例 6-4】** 用复位和置位指令编写例 6-2 的程序。

**解：** 梯形图如图 6-34 和图 6-35 所示。

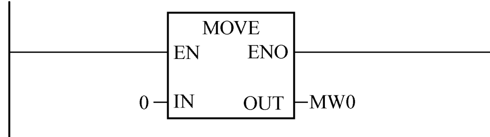
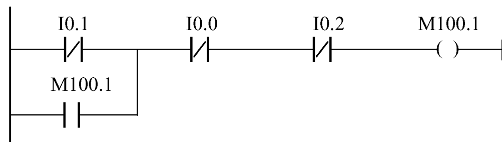
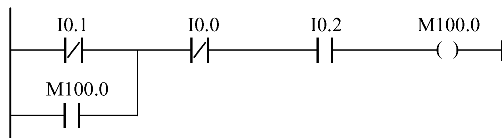
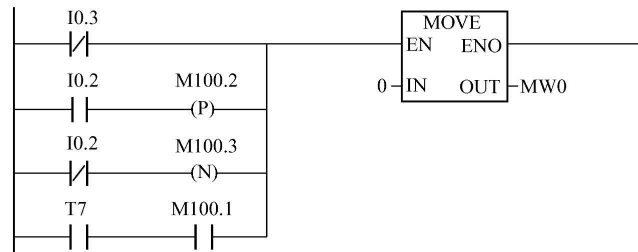



图 6-34　OB100 中的程序



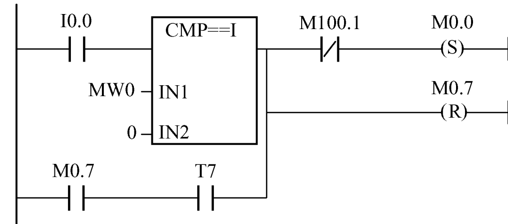
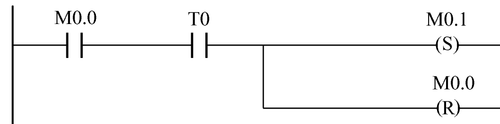
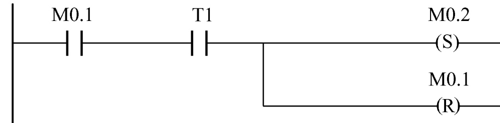




图 6-35　OB1 中的程序

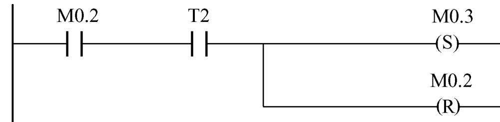
程序段5：标题



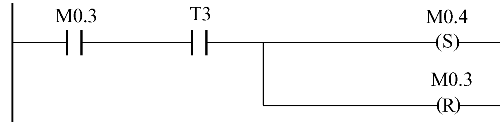

程序段6：标题



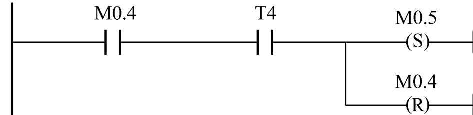

程序段7：标题



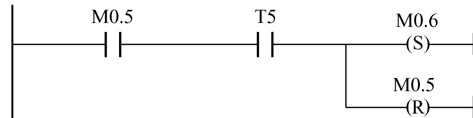

程序段8：标题



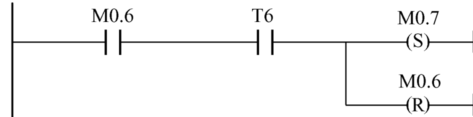

程序段9：标题：



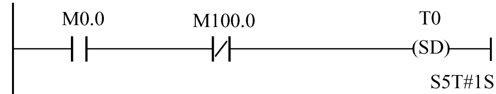

程序段10：标题：




程序段11：标题：




程序段12：标题：




程序段13：标题：

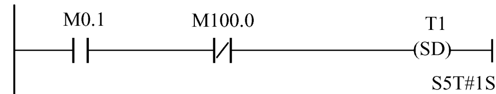
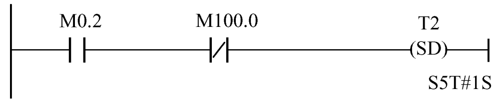
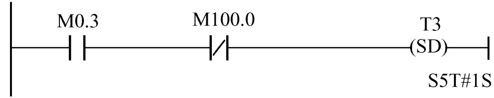



图 6-35　OB1 中的程序（续）

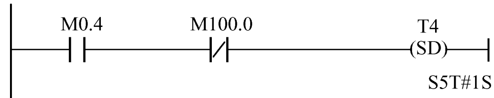
程序段14：标题：



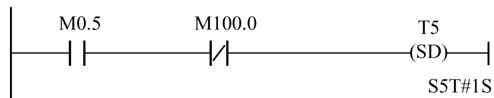

程序段15：标题：



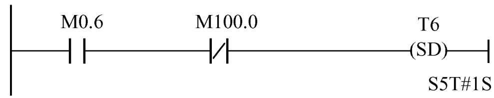

程序段16：标题：



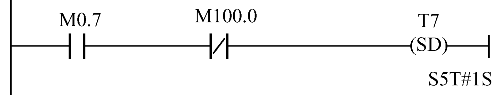

程序段17：标题：




程序段18：标题：




程序段19：标题：




程序段20：标题：




程序段21：标题：

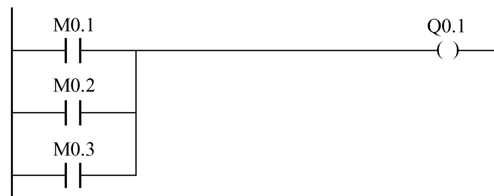



图 6-35　OB1 中的程序（续）

程序段22：标题：




程序段23：标题：




图 6-35　OB1 中的程序（续）

至此，同一个顺序控制的问题使用了基本指令、复位和置位指令和功能指令，共三种解决方案编写程序。三种解决方案的编程都有各自几乎固定的步骤，但有一步是相同的，那就是首先都要设计功能图。3 种解决方案没有优劣之分，读者可以根据自己的实际情况选用。

## 6.3　S7-300/400 PLC 的调试方法

### 6.3.1　用变量监控表进行调试

变量表是很有用的调试工具，但不能下载到 PLC 中，因此从 PLC 中上传的程序没有变量表。

**1. 变量表的功能**

变量表和 PLC 建立在线联系后，可以将硬件组态和程序下载到 PLC 中。用户可以通过 STEP 7 进行在线调试程序，寻找并发现程序设计中的问题。变量表上可以显示用户指定的变量，它可以用于监视和修改变量值。变量表有如下功能：

1）监视变量。可以在编程设备上显示用户程序或 CPU 中每个变量值的当前值。

2）修改变量。可以将固定值赋给用户程序或 CPU 中的每个变量，使用程序状态测试时进行一次数值修改。

3）使用外部设备输出并激活修改值，允许在停机状态下将固定值赋给 CPU 的 I/O。

4）强制变量。可以为用户程序或 CPU 中的每个变量赋予一个固定值，这个值是不能被用户程序覆盖的。

用户可以显示或者赋值的变量包括：输入、输出、位存储、定时器、计数器、数据块的内容和 I/O。

**2. 建立变量表**

在 SIMATIC 管理器界面中，如图 6-36 所示。先选中“块”，再单击菜单栏中的“插

入”→“S7 块”→“变量表”，弹出“属性－变量”界面，如图6-37 所示，默认变量表为“VAT_1”，单击“确定”按钮，便可建立变量表。




图 6-36　建立变量表




图 6-37　“属性－变量表”界面

在 SIMATIC 管理器界面中，双击“VAT_1”，弹出变量表界面，此时的变量表是空白的，并没有变量，如图6-38 所示。




图 6-38　打开变量表

**3. 利用变量表调试程序**

（1）输入变量

每个变量表中有 5 个栏，分别显示变量的 5 个属性：地址、符号、显示格式、状态值和修改值。一个变量表最多有 1024 行，每行最多可有 255 个字符。

用户可以通过在“符号”栏输入符号或在“地址”栏输入地址来插入变量，如果在符号表中已经定义地址相应的符号，则符号栏或者地址会自动输入，如图 6-39 所示。

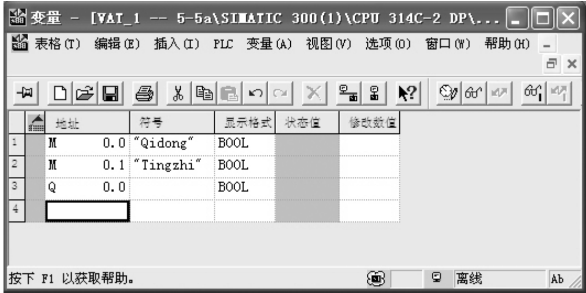



图 6-39　输入变量

（2）监视和修改变量

假设要调试的程序如图 6-40 所示，将整个项目下载到 CPU 中，注意变量表不能下载到 CPU 中去。

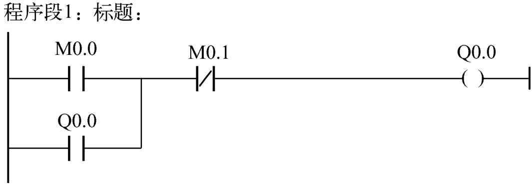



图 6-40　程序

1）变量表与 CPU 的连接。先建立变量表与 CPU 的连接，共有三种方法。单击菜单中的“PLC”→“连接到”来建立变量表与 CPU 的连接。子菜单有 3 个选项，如图 6-41 所示，第一个是组态的 CPU，其作用与单击工具栏中的按钮作用相同，用于建立被激活的变量表与 CPU 的连接；第二个是直接连接 CPU，其作用与单击工具栏中的按钮作用相同，用于直接连接 CPU（与编程设备用编程电缆连接的 CPU）之间的在线的连接；第三个是可访问的 CPU，在打开的对话框中，用户可以选择与哪个 CPU 建立连接。




图 6-41　变量表与 CPU 的连接

使用菜单命令“PLC”→“断开连接”，可以断开变量表与 CPU 的连接。

2）变量表的监视。单击工具栏中的“监视变量”按钮，或者使用菜单中的“变量”→“监视”，便可监视程序中变量的情况，如图 6-42 所示，3 个变量的状态都有显示。



| | 地址 | | 符号 | 显示格式 | 状态值 | 修改数值 |
|---|---|---|---|---|---|---|
| 1 | M | 0.0 | "Qidong" | BOOL | true | true |
| 2 | M | 0.1 | "Tingzhi" | BOOL | false | false |
| 3 | Q | 0.0 | | BOOL | true | |
| 4 | | | | | | |

图 6-42　变量表的监视和修改

3）变量表的修改。当变量表处于监视状态时，在参数“M0.0”的“修改数值”栏中输入“true”（1 也可以），再单击“修改变量”按钮，可以看到参数“M0.0”为“true”，由于程序运行使得参数“Q0.0”也为“true”。当然也可以使用菜单中的“变量”→“修改”，来修改参数的数值。

**【关键点】** 在仿真器中“修改变量”时，仿真器必须置于“RUN－P”模式下，而真实的 S7－300/400 PLC（2002 年后生产新型号）无“RUN－P”模式。仿真器监视参数时，置于“RUN”或者“RUN－P”模式均可。

（3）强制变量

强制变量可以给用户一个固定值，它独立于程序运行，不会被执行的用户程序改变或者覆盖。强制的优点在于可以在不改变程序代码，也不改变硬件连线的情况下，强行改变输入和输出状态。

强制变量的方法是：先选中要强制变量中将要修改的数值，再使用菜单中的“变量”→“显示强制值”，如图 6-43 所示。在需要强制的变量后输入强制值，本例强制 I0.0 为“true”，选中强制值，单击鼠标右键，弹出快捷菜单，单击“强制”，则 I0.0 的常开触点闭合。变量中的 I0.0 前面一个红色“F”标记，如图 6-44 所示。

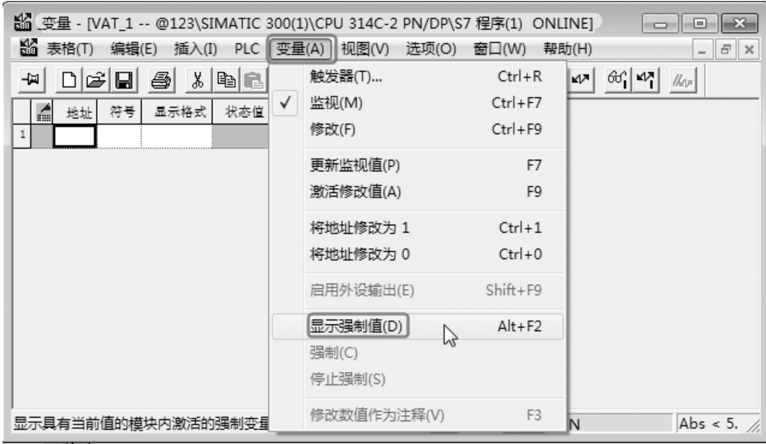



图 6-43　“显示”强制




图 6-44　强制

停止强制的方法是：使用菜单中的“变量”→“删除强制”，如图 6-43 所示。

**【关键点】**

1）利用“修改变量”功能可以同时输入几个数据。“修改变量”的作用类似于“强制”的作用。但两者是有区别的。

2）强制功能的优先级别要高于“修改变量”，“修改变量”的数据可能改变参数状态，但当与逻辑运算的结果抵触时，写入的数值也可能不起作用。

3）修改变量不能改变输入继电器（如 I0.0）的状态，而强制可以改变。

4）仿真器中可以模拟“修改变量”，但不能模拟“强制”功能，强制功能只能在真实的 S7－300/400 PLC 中实现。

5）此外，PLC 处于强制状态时，“FRCE”灯为黄色，正常运行状态时，不应使 PLC 处于强制状态，强制功能仅用于调试。

**【例 6-5】** 如图 6-45 所示的梯形图，Q0.0 状态为 1，问在“状态表”中，分别用“修改变量”“强制”功能，是否能将 Q0.0 的数值变成 0？

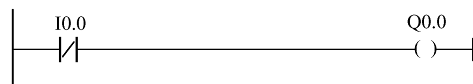



图 6-45　梯形图

**解：**用“修改变量”功能不能将 Q0.0 的数值变成 0，因为图 6-45 的梯形图的逻辑运算的结果造成 Q0.0 为 1，与“修改”结果抵触，最后输出结果以逻辑运算的结果为准。

用“强制”功能将 Q0.0 的数值变成 0，因为强制的结果可以覆盖逻辑运算的结果。

## 6.3.2　使用 PLCSIM 软件进行调试

### 1. S7－PLCSIM 简介

西门子为 S7－300/400 PLC 设计了一款可选仿真软件包 PLC Simulation（本书简称 S7－PLCSIM），此仿真软件包可以在计算机或者编程设备中模拟可编程序控制器运行和测试程序，它不能脱离 STEP 7 独立运行。如果 STEP 7 中已经安装仿真软件包，工具栏中的“仿真开关”按钮是亮色的，否则是灰色的，只有“仿真开关”按钮是亮色才可以用于仿真。

S7－PLCSIM 提供了简单的用户界面，用于监视和修改在程序中使用各种参数（如开关量输入和开关量输出）。当程序由 S7－PLCSIM 处理时，也可以在 STEP 7 软件中使用各种软件功能，如使用变量表监视、修改变量和断点测试功能。

**2. S7－PLCSIM 应用**

S7－PLCSIM 仿真软件使用比较简单，以下用一个简单的例子介绍其使用方法。

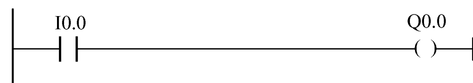



图 6-46　用于仿真的程序

**【例 6-6】** 将如图 6-46 所示的程序用 S7－PLCSIM 进行仿真。

**解：** 具体步骤如下：

1）新建一个项目，并进行硬件组态，在组织块 OB1 中输入如图 6-46 所示的程序，保存项目。

2）开启仿真。在 SIMATIC 管理界面中，单击工具栏上的“仿真开关”按钮，如图 6-47 所示。

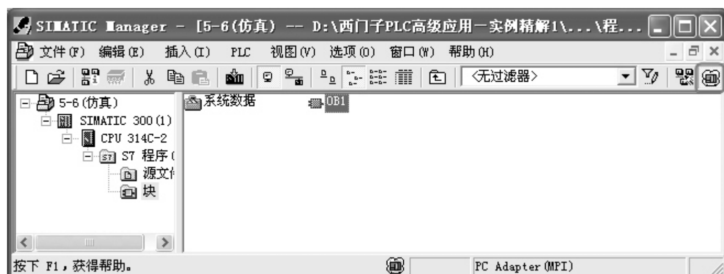



图 6-47　开启仿真

3）下载程序。先选定“SIMATIC 300（1）”，再单击工具栏的“下载”按钮，将硬件组态和程序下载到仿真器中，如图 6-48 所示。




图 6-48　下载程序

4）进行仿真。先选择“RUN”，也就是将仿真器置于运行状态，再在 I0.0（IB0 的第 0 位）上选取“√”，也就是将 I0.0 置于“ON”，这时 Q0.0 也显示为“ON”；当去掉 I0.0 上的“√”，也就是将 I0.0 置于“OFF”，这时 Q0.0 上的“√”消失，即显示为“OFF”，如图 6-49 所示。

5）监视运行。打开程序编辑器，在工具栏中单击“监视”按钮，可以看到：若仿真器上的 I0.0 和 Q0.0 都是“ON”，则程序编辑器界面上的 I0.0 和 Q0.0 也都是“ON”，如图 6-50 所示。这个简单例子的仿真效果与下载程序到 PLC 中的效果基本相同，相比之下前者实施要容易得多。

**3. S7－PLCSIM 与真实 PLC 的差别**

S7－PLCSIM 提供了方便、强大的仿真模拟功能。与真实的 PLC 相比，它的灵活性高，

提供了许多 PLC 硬件无法实现的功能，使用也更加方便。但是仿真软件毕竟不能完全取代真实的硬件，不可能实现完全仿真。用户利用 S7 - PLCSIM 进行仿真时，还应该了解它与真实 PLC 的差别。

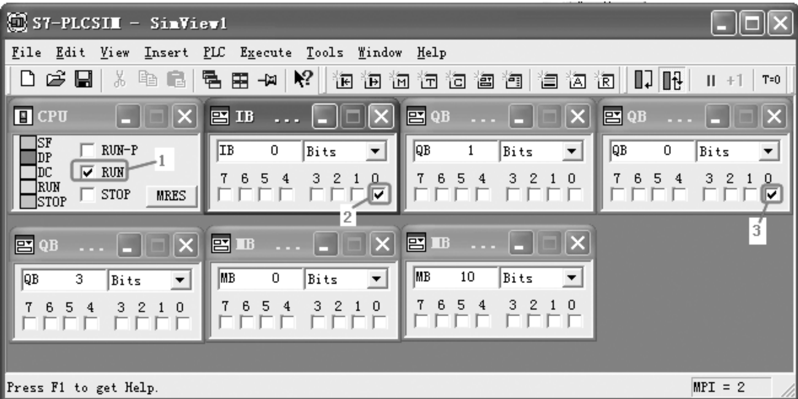



图 6-49　进行仿真




图 6-50　监视运行

（1）S7 - PLCSIM 上有如下功能在真实的 PLC 上无法实现

1）仿真的 CPU 中正在运行时可以用“Stop”选项中断程序，恢复“运行”时是从程序中断处开始继续处理程序。

2）与真实的 CPU 一样，仿真软件可以改变 CPU 的操作模式（RUN、RUN - P 和 STOP）。但与实际 CPU 不同的是仿真的 CPU 切换到 STOP 模式并不会改变输出的状态。

3）仿真软件中在目标视图中变量的每个改变，其存储区对应相关地址的内容会被同时更新。CPU 并不是等到循环周期结束或开始时才更新改变的数据。

4）使用关于程序处理的选项可以指定 CPU 如何执行程序：

- 选择“By cycles”程序执行一个周期后等待命令再执行下一个循环周期。
- 选择“Automatic”程序的处理同实际自控系统一样，一旦一个循环周期结束马上执行下一个周期。

5）仿真定时器可以使用自动或手动方式处理，自动方式按照程序执行结果，手动方式可以给定特殊值或复位定时器。复位定时器可以复位单独的定时器或一次复位所有定时器。

6）可以手动触发诊断中断 OB。OB40 ~ OB47（过程中断）、OB70（I/O 冗余错误）、

OB72（CPU 冗余错误）、OB73（通信冗余错误）、OB80（时间错误）、OB82（诊断警告）、OB83（插拔模块警告）、OB85（程序执行错误）和 OB86（机架故障）。

7）过程映像区和 I/O 区。如果改变一个输入映像区的值，S7 – PLCSIM 立即将此值复制到输入外设区。这就意味着从输入外设区写到输入过程映像区所需要的值在下一个循环周期开始时不会丢失。同样如果改变了输出映像区的一个值，此值立即被复制到输出外设区。

（2）S7 – PLCSIM 与"实际"的自动化系统还有以下不同

1）诊断缓冲区。S7 – PLCSIM 不能支持所有写入诊断缓冲区的错误消息。例如，关于 CPU 中的电池电量不足的消息或者 EEPROM 错误是不能仿真的。但大部分 I/O 和程序错误都是可以仿真的。

2）在改变操作模式时（比如从 RUN 切换到 STOP）输入/输出没有"安全"状态。

3）不支持功能模块（FM）。

4）S7 – PLCSIM 与 S7 – 400 CPU 一样支持 4 个累加器。在某些情况下 S7 – PLCSIM 上运行的程序与真实的只有两个累加器，与 S7 – 300 CPU 上运行结果不同。

5）输入/输出的不同。大多数 S7 – 300 PLC 产品系列的 CPU 可以自动配置输入/输出设备。如果将模块连接到控制器，CPU 即自动识别此模块。对于仿真的自动化系统，这种自动识别是不能模拟的。如果把一个自动组态好 I/O 的 S7 – 300 CPU 程序装载到 S7 – PLCSIM 中，系统数据中将不包含任何 I/O 组态。因此，如果使用 S7 – PLCSIM 来仿真 S7 – 300 的程序，为了使 CPU 能识别所使用的模块，必须首先装载硬件组态。在 S7 – PLCSIM 中，S7 – 300 CPU 不能自动识别 I/O，例如 CPU 315 – 2DP、CPU 316 – 2DP 或 CPU 318 – 2DP 等，为了能将硬件组态装载到 S7 – PLCSIM，需要创建一个项目。复制相应的硬件组态到这个项目并装载到 S7 – PLCSIM。然后从任意 STEP 7 项目装载程序块，I/O 处理都不会有错误。

此外，S7 – PLCSIM V5.4 SP3 以前的版本不能对通信进行仿真。

### 6.3.3 使用"断点"调试程序

**1. "断点"调试简介**

断点调试是调试程序的一种常用方法，C 语言等高级语言支持断点调试功能，西门子的 S7 – 300/400 PLC 支持这种调试功能。

断点调试是指在程序的某一行设置一个断点，调试时，程序运行到这一行就停下来，然后可以一步一步往下调试，调试过程中可以观察各个变量当前的值，如出现错误，调试到出错的代码行即显示错误，停下。编程者借助显示的异常代码或者数值，找出程序中的错误。

**2. "断点"调试实施方法**

以下用一个简单的例子说明断点调试的过程，简单的调试程序如下：

```
A     I      0.0
      JNB    _001
      L      MW      0
      T      MW      2
      SET
      SAVE
      CLR
_001: A      BR
      =      Q       0.0
```

1）在程序编辑器界面的菜单栏中，单击“调试”→“操作”，弹出如图 6-51 所示的界面，选中“测试操作”选项，单击“确定”按钮。



图 6-51　操作

2）在程序编辑器界面的菜单栏中，单击“视图”→“断点栏”，调出断点调试工具栏，此时的断点调试工具栏是灰色。单击工具栏上的“打开在线/离线”按钮，确保 STEP 7 与 PLC 处于“连接”状态，此时可以看到：断点工具栏是亮色，表明可以使用该工具栏了。

3）单击断点工具栏上的“断点激活（开/关）”按钮，将程序切换到指令表状态，选中“1 处”，单击断点工具栏的“设置/删除断点”按钮，完成设置断点，如图 6-52 和图 6-53 所示。

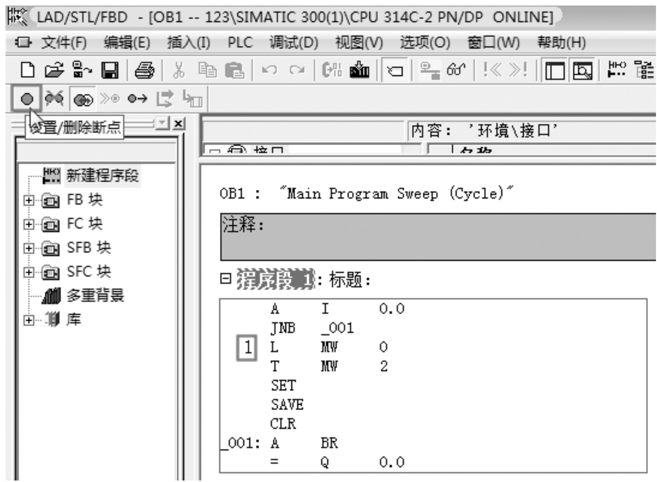


图 6-52　设置断点（1）

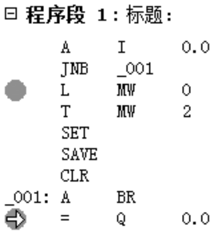


图 6-53　设置断点（2）

4）单击断点工具栏的“下一语句”按钮一次，则程序执行停止在如图 6-53 所示的两个断点中的其中一个断点的位置，并显示在此位置的当前值。此时 CPU 处于“HOLD”状态。

### 6.3.4　使用参考数据调试程序

参考数据包括交叉参考、输入/输出和位存取区（I/Q/M）分配列表、程序结构、未使用的符号和不带符号的地址视图，以下分别介绍。

**1. 交叉参考**

交叉参考能显示程序中元件使用的详细信息。交叉参考对查找程序中数据地址的使用十分有用。在程序编辑器中，单击菜单栏中的“选项”→“参考数据”→“显示”，如图 6-54 所示，选择图 6-55 中的“交叉参考”选项，单击“确定”按钮，弹出“交叉参考”界面，如图 6-56 所示，所有参数和内部继电器的具体位置都列在表格中（在哪个块的哪个程序段），这个功能十分有用，便于编程者搜索参数的具体位置（定位功能）。

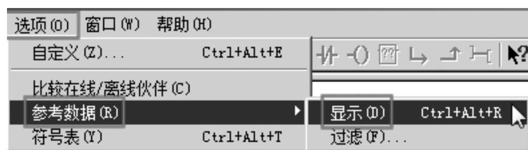



图 6-54　打开交叉参考（1）

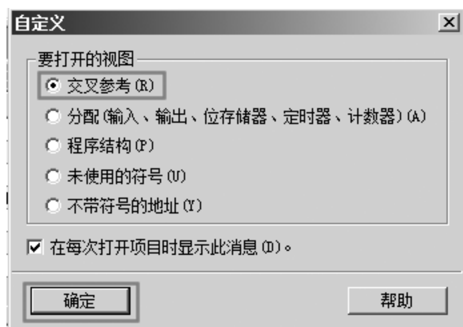



图 6-55　打开交叉参考（2）

参考 - [S7 程序(1) (交叉参考) -- OB100\SIMATIC 300(1)\CPU 314C-2 DP]

参考数据(R)　编辑(E)　视图(V)　窗口(W)　帮助(H)

已经过滤

| 地址(符号) | 块(符号) | 类型 | 语言 | 位置 | 位置 |
|---|---|---|---|---|---|
| I 0.2 | OB1 | R | LAD | NW 1 /A | |
| M 10.0 | OB1 | R | LAD | NW 2 /A | |
| MB 10 | OB1 | W | LAD | NW 3 /T | |
| Q 0.0 | OB100 (COMPLETE RESTART) | W | LAD | NW 2 /S | NW 3 /R |
| Q 0.7 | OB1 | W | LAD | NW 1 /= | |
| Q 2.0 | OB1 | W | LAD | NW 2 /= | |
| QB 0 | OB1 | R | LAD | NW 3 /L | |

将显示过滤后的数据。　NUM

图 6-56　交叉参考

**2. 输入/输出和位存取区（I/Q/M）分配列表**

单击工具栏中的“输入/输出、位存储器、定时器和计数器的赋值”▦按钮，弹出如图 6-57 所示的界面，从图中可以看出，程序中 Q0.0、Q0.7 和 QB0 都用到了，而 QB0 包含 Q0.0、Q0.7，所以编写完程序后，在交叉参考中看到这种现象时，一定要分析 QB0 与 Q0.0、Q0.7 是否冲突，这个功能十分有用。

参考 - [S7 程序(1) (赋值) -- OB100\SIMATIC 300(1)\CPU 314C-2 DP]

参考数据(R)　编辑(E)　视图(V)　窗口(W)　帮助(H)

无过滤

输入，输出，位存储器

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | B W D |
|---|---|---|---|---|---|---|---|---|---|
| IB 0 | | | | | | X | | | |
| QB 0 | X | | | | | | | X | T |
| QB 1 | | | | | | | | | |
| QB 2 | | | | | | | | X | |
| MB10 | | | | | | | | X | T |

定时器，计数器

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| T0-9 | | | | | | | |
| C0-9 | | | | | | | |

按 F1 获得帮助。　NUM

图 6-57　输入/输出和位存取区（I/Q/M）分配列表

**3. 程序结构**

程序结构反映了用户程序内的块的调用体系以及所使用块及其嵌套等级的概况。单击工具栏中的“程序结构”按钮☰，弹出如图 6-58 所示的界面。

**4. 符号表的输出**

对于一个大型的项目，一定要用到“符号寻址”，这样必然要用到符号表，打开符号编

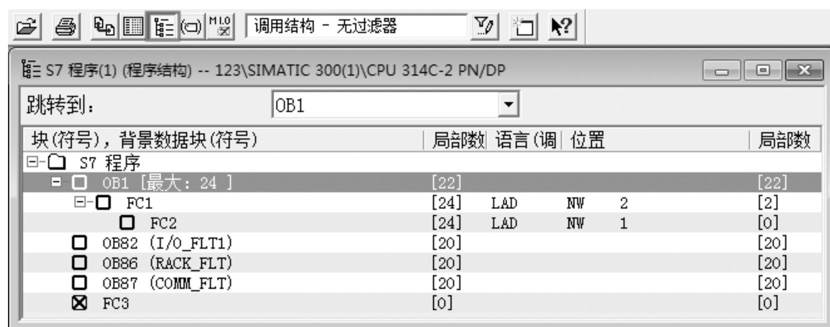



图 6-58　程序结构

辑器，如图 6-59 所示。单击“符号编辑器”菜单栏“符号表”→“输出”，输入如图 6-60 所示的文件名，单击“确定”按钮即可。注意这个保存的文件是 ASCII 码格式，可以用 Excel 软件打开，如图 6-61 所示。




图 6-59　符号表




图 6-60　输出文件

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 126,COMPL | RESTART | OB | 100 | OB | 100 | Complete | Restart | |
| 2 | 126,Start | | I | 0 | BOOL | | 启动 | | |
| 3 | 126,Stop | | I | 0.1 | BOOL | | 停止 | | |
| 4 | 126,Motor | | Q | 0 | BOOL | | 电动机 | | |
| 5 | | | | | | | | | |
| 6 | | | | | | | | | |

图 6-61　用 Excel 软件打开

**5. 符号表的导入**

在符号编辑器中，单击符号编辑器菜单栏“符号表”→“导入”，选中以前导的以前输出的文件“1. ASC”，则可以将“1. ASC”导入到参数表。

## 6.4 实例

初学者在进行 PLC 控制系统的设计时，往往不知从何着手，其实 PLC 控制系统的设计有一个相对固定的模式，只要读者掌握了前述章节的知识，再按照这个模式进行，一般不难设计出正确的控制系统和程序。以下通过两个例子来说明 PLC 控制系统的设计过程。

**【例 6-7】**液体混合装置示意图如图 6-62 所示，上限位、下限位和中限位液位传感器被液体淹没时为 1 状态，电磁阀 A、B 和 C 的线圈通电时，阀门打开，电磁阀 A、B 和 C 的线圈断电时，阀门关闭。在初始状态时容器是空的，各阀门均关闭，各传感器均为 0 状态。按下起动按钮后，打开电磁阀 A，液体 A 流入容器，中限位开关变为 ON 时，关闭 A，打开阀 B，液体 B 流入容器。液面上升到上限位开关，关闭阀门 B，电动机 M 开始运行，搅拌液体，30 s 后停止搅动，打开电磁阀 C，放出混合液体，当液面下降到下限位开关之后，过 3 s，容器放空，关闭电磁阀 C，打开电磁阀 A，又开始下一个周期的操作。按停止按钮，当前工作周期结束后，才能停止工作，按急停按钮可立即停止工作。请绘制功能图，设计梯形图。

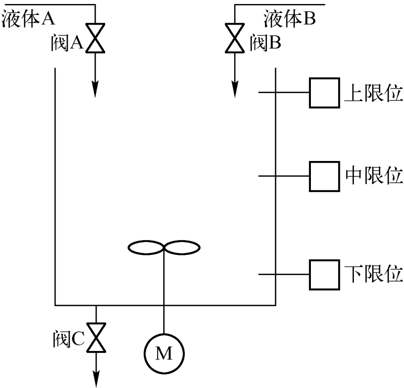


图 6-62　液体混合装置

**解：**液体混合装置 PLC 的 I/O 分配见表 6-1。

**表 6-1　PLC 的 I/O 分配表**

| 输入 | | | 输出 | | |
|---|---|---|---|---|---|
| 名称 | 符号 | 输入点 | 名称 | 符号 | 输出点 |
| 开始按钮 | SB1 | I0. 0 | 电磁阀 A | YA1 | Q0. 0 |
| 停止按钮 | SB2 | I0. 1 | 电磁阀 B | YA2 | Q0. 1 |
| 急停 | SB3 | I0. 2 | 电磁阀 C | YA3 | Q0. 2 |
| 上限位传感器 | SQ1 | I0. 3 | 电动机 | M | Q0. 3 |
| 中限位传感器 | SQ2 | I0. 4 | | | |
| 下限位传感器 | SQ3 | I0. 5 | | | |

电气系统的原理图如图 6-63 所示，功能图如图 6-64 所示，梯形图如图 6-65 所示。

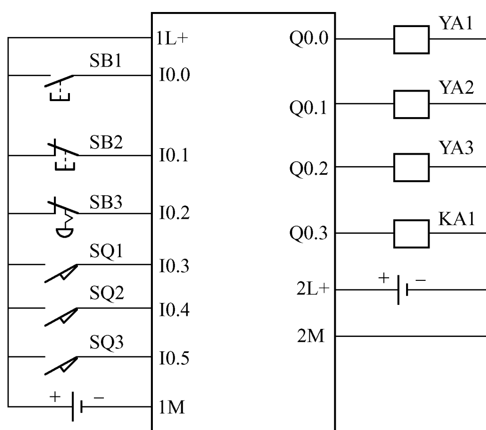



图 6-63　原理图

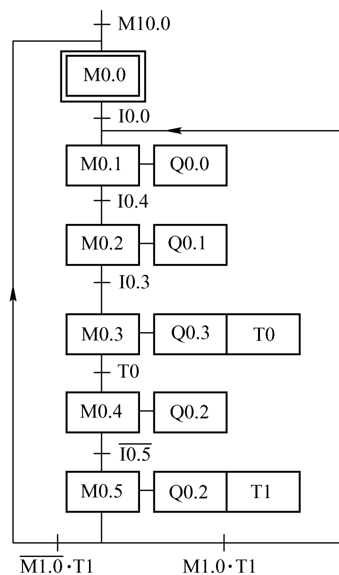
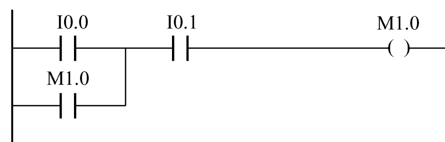
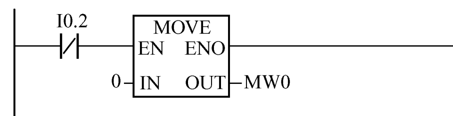



图 6-64　功能图

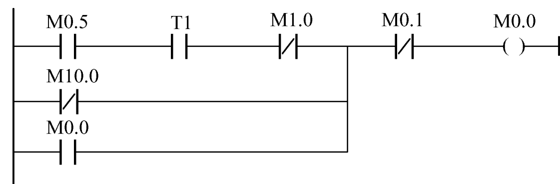
程序段1：连续




程序段2：急停




程序段3：标题




程序段4：标题

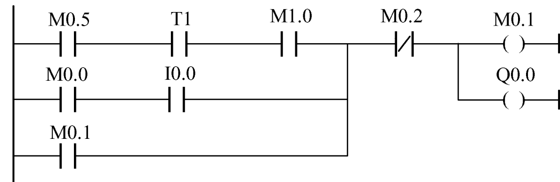
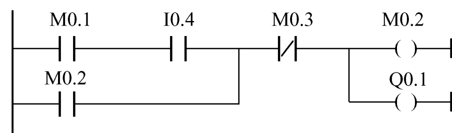
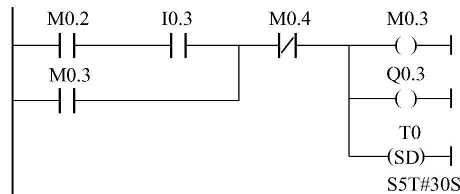



图 6-65　梯形图

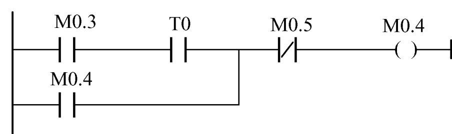
程序段5：标题



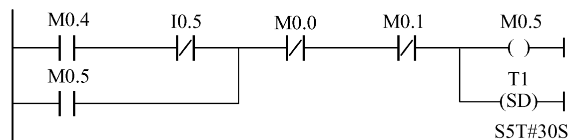

程序段6：标题



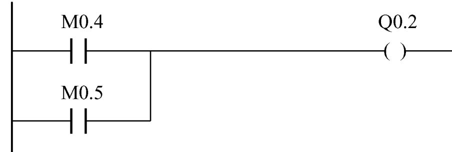

程序段7：标题




程序段8：标题




程序段9：标题：




程序段10：标题：

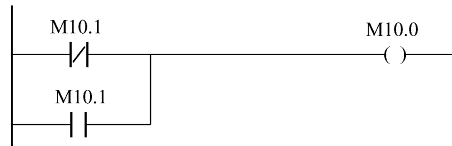



图 6-65　梯形图（续）

**【例 6-8】** 某钻床用两个钻头同时钻两个孔，开始自动运行之前，两个钻头在最上面，上限位开关 $SQ_1$ 和 $SQ_3$ 为“ON”。操作人员放好工件，按起动按钮 I0.0 后。工件被夹紧后，两个钻头同时开始工作，钻到由限位开关 $SQ_2$ 和 $SQ_4$ 设定的深度时分别上行，回到由限位开关 $SQ_1$ 和 $SQ_3$ 设定的起始位置时，分别停止上行。当两个钻头都到起始位置后，工件松开，工件松开后，加工结束，系统回到初始状态。钻床的加工示意图如图 6-66 所示，请设计功能

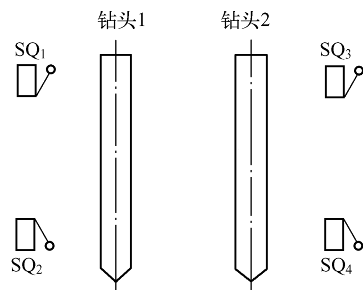



图 6-66　钻床加工示意图

图和梯形图。

**解**：钻床的 PLC 的 I/O 分配见表 6-2。

**表 6-2　PLC 的 I/O 分配表**

| 输　入 | | | 输　出 | | |
|---|---|---|---|---|---|
| 名　称 | 符　号 | 输　入　点 | 名　称 | 符　号 | 输　出　点 |
| 开始按钮 | SB1 | I0. 0 | 夹具夹紧 | KA1 | Q0. 0 |
| 停止按钮 | SB2 | I0. 1 | 钻头 1 下降 | KA2 | Q0. 1 |
| 钻头 1 上限位开关 | SQ1 | I0. 2 | 钻头 1 上升 | KA3 | Q0. 2 |
| 钻头 1 下限位开关 | SQ2 | I0. 3 | 钻头 2 下降 | KA4 | Q0. 3 |
| 钻头 2 上限位开关 | SQ3 | I0. 4 | 钻头 2 上升 | KA5 | Q0. 4 |
| 钻头 2 下限位开关 | SQ4 | I0. 5 | 夹具松开 | KA6 | Q0. 5 |
| 夹紧限位开关 | SQ5 | I0. 6 | | | |
| 松开下限位开关 | SQ6 | I0. 7 | | | |

电气系统的原理图如图 6-67 所示，功能图如图 6-68 所示，梯形图如图 6-69 所示。

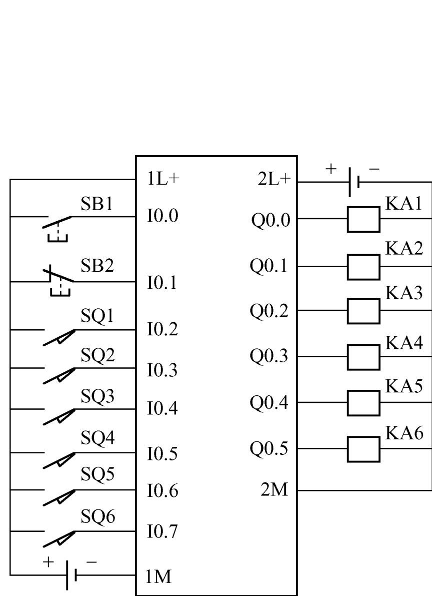



图 6-67　原理图

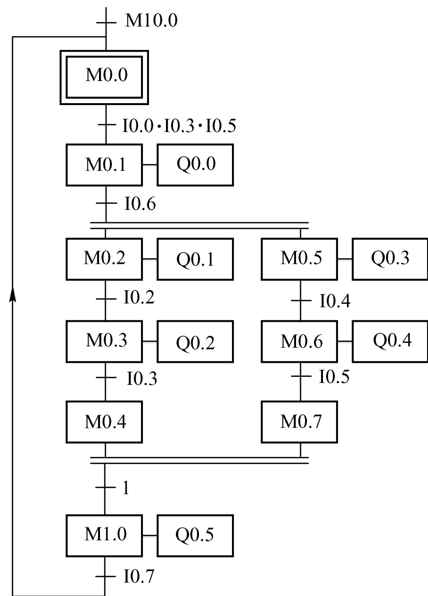



图 6-68　功能图

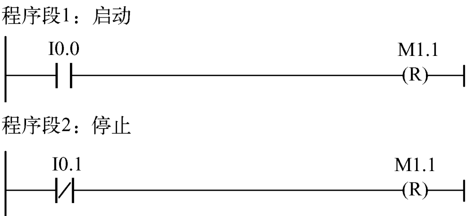
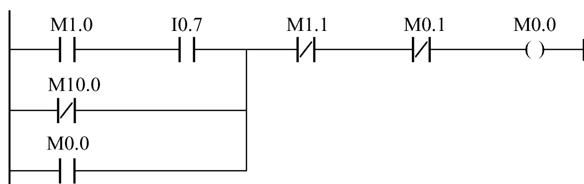
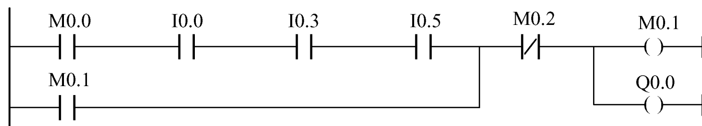



图 6-69　梯形图

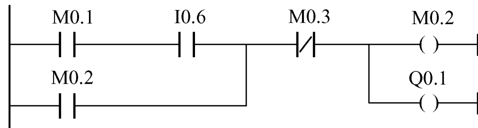
程序段3：标题：



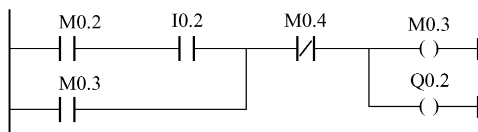

程序段4：标题：



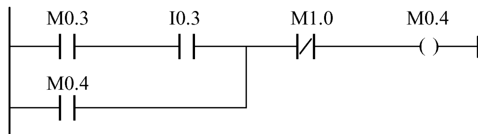

程序段5：标题：



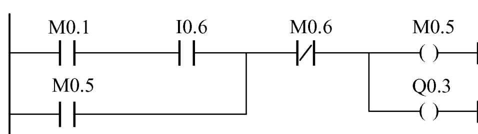

程序段6：标题：



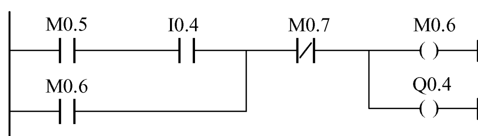

程序段7：标题：




程序段8：标题：




程序段9：标题：




程序段10：标题：

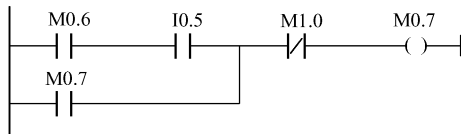
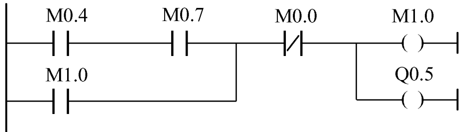



图 6-69　梯形图（续）

程序段11：标题：




程序段12：确保M10.0首次扫描接通一个扫描周期

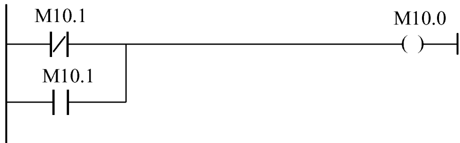



图 6-69　梯形图（续）

# 第 7 章　西门子 PLC 的 SCL 和 GRAPH 编程

本章介绍 S7 – SCL 和 S7 – GRAPH 的软件安装、应用场合和语言特点等，并最终使读者掌握 S7 – SCL 和 S7 – GRAPH 的程序编写方法。

## 7.1　西门子 PLC 的 SCL 编程

### 7.1.1　S7 – SCL 简介

**1. S7 – SCL 概念**

S7 – SCL（Structured Control Language）结构化控制语言是一种类似于计算机高级语言的编程方式，它的语法规范接近计算机中的 Pascal 语言。

**2. S7 – SCL 特点**

1）它符合国际标准 IEC 61131 – 3。

2）获得了 PLCopen 基础级认证。

3）一种类似于 Pascal 的高级编程语言。

4）适用于 SIMATIC S7 – 300（推荐用于 CPU314 以上 CPU）、S7 – 400、C7 和 WinAC 产品。S7 – SCL 为 PLC 做了优化处理，它不仅仅具有 PLC 典型的元素（例如 输入/输出，定时器、计数器及符号表），而且具有高级语言的特性，例如：循环、选择、分支、数组和高级函数。

5）S7 – SCL 可以编译成 STL，虽然其代码量相对于 STL 编程有所增加，但程序结构和程序的总体效率提高了。类似于计算机行业的发展，汇编语言已经被舍弃，取而代之的是 C/C ++ 等高级语言。S7 – SCL 对工程设计人员要求较高，需要其具有一定的计算机高级语言的知识和编程技巧。

**3. S7 – SCL 应用范围**

由于 S7 – SCL 是高级语言，所以其非常适合于如下任务：

1）复杂运算功能。

2）复杂数学函数。

3）数据管理。

4）过程优化。

### 7.1.2　S7 – SCL 的软件包安装

西门子的 STEP 7 的软件包中的标准配置中只有 LAD、STL 和 FBD 三种编程方式，要使用 S7 – SCL 编写程序，需要安装 S7 – SCL 软件包，安装此软件包需要购买授权。

**1. S7 – SCL 与操作系统的兼容性**

不同的 S7 – SCL 版本与操作系统有兼容性问题，简单说如果 S7 – SCL 的版本较低（早

期的版本）可能无法安装到较新的操作系统上，例如 S7 - SCL V5.3 SP1 不能安装在 Win2008 R2（64）操作系统中。S7 - SCL 与操作系统的兼容性列表见表 7-1。

**表 7-1　S7 - SCL 与操作系统的兼容性**

| 产品版本 | STEP7 V5.5 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | WinXP sp2 | WinXP sp3 | Win7 | Win7 sp1 | Win7（64） | Win7 sp1（64） | Win2008 R2（64） | Win2008 R2 SP1（64） |
| S7 - SCL V5.3 | X | - | - | - | - | - | - | - |
| S7 - SCL V5.3 SP5 | X | X | X | - | - | - | - | - |
| S7 - SCL V5.3 SP6 | X | X | X | X | X | X | X | X |

注：1. 表中的“X”表示兼容。
2. “-”表示不兼容。
3. 举例：Win7（64）操作系统需要安装 S7 - SCL V5.3 SP6 以上版本的软件包。

**2. 安装 S7 - SCL 注意事项**

与安装 STEP 7 相同。

### 7.1.3　S7 - SCL 源文件编辑器

**1. 打开 SCL 源文件编辑器**

新建 SCL 源文件，选择“源文件”，单击鼠标右键，选择“插入对象”→“SCL 源文件”，如图 7-1 所示，此时，插入了“SCL 源文件（1）”，双击“SCL 源文件（1）”，弹出 SCL 源文件编辑器，如图 7-2 所示。




图 7-1　打开 SCL 源文件编辑器

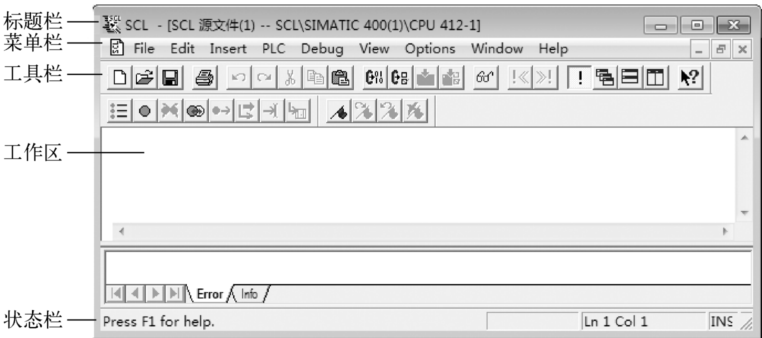



图 7-2　SCL 源文件编辑器

**2. SCL 源文件编辑器的界面介绍**

如图 7-2 所示，SCL 源文件编辑器的界面分 5 个区域，以下分别介绍。

① 标题栏：包含窗口的标题（本例为 SCL）和窗口的控制按钮（如最大化、退出窗口等按钮）。

② 菜单栏：包含打开窗口的所有菜单，如 File（文件）、Edit（编辑）及 Insert（插入）等。

③ 工具栏：包含使用较频繁的菜单命令的按钮。工具栏简化了操作。

④ 工作区：包含一个或者多个窗口，这些窗口用于编写程序、调试和显示编译信息。这是最为重要的区域。

⑤ 状态栏：主要显示激活状态对象的状态。

**3. SCL 源文件编辑器的菜单**

SCL 源文件编辑器的菜单有 File（文件）、Edit（编辑）、Insert（插入）、PLC、Debug（调试）、View（视图）、Option（选项）、Windows（窗口）和 Help（帮助）。常用的菜单也配有工具条。

（1）File（文件）

菜单中的 File（文件）中主要包括 New（新建文件）、Open（打开文件）、Close（关闭打开的文件）、Save（保存文件）和 Compile（编译文件）等。前面的四项与 Office 软件的此菜单含义一样，在此不赘述。Compile（编译文件）的作用是程序编写完成后，起编译作用，如程序有错误，将在窗口显示。类似于 STEP 7 中的编译功能。

（2）Edit（编辑）

菜单中的 Edit（编辑）中主要包括 Undo（撤销）、Redo（重复）、Cut（剪切）、Copy（复制）、Paste（粘贴）和 Select All（全部选择）等。其使用方法和 Office 软件的此菜单含义一样，在此不赘述。

（3）Insert（插入）

菜单中的 Insert（插入）包括 Block Call（块调用）、Block Template（块模板）和 Control Structure（控制结构）。块调用主要用于调用 STEP 7 的库中已经安装的块（如 FC105）；块模板主要用于生成一些标准的模板（如 FC、FB 等），使用模板可简化程序编写；控制结构中有一些常用语法结构，使用控制结构可减少书写程序量。

（4）PLC

菜单中的 PLC 选项主要包含 Download（下载）、Select Online CPU（选择在线 CPU）和 Monitor/Modify Variable（监控/修改变量表）等选项。

（5）Debug（调试）

菜单中的 Debug（调试）包括 Operation（操作模式）、Monitor（监控）和 Set Breakpoint（设置断点）等。含义类似于 STEP 7 中的调试功能。

## 7.1.4 S7-SCL 编程语言

**1. S7-SCL 的基本术语**

（1）字符集

S7-SCL 使用 ASCII 字符子集：字母 A～Z（大小写）、数字 0～9 以及空格和换行符等。此外，还包含特殊含义的字符，见表 7-2。

表 7-2　S7-SCL 的特殊含义字符

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| + | - | * | / | = | < | > | [ | ] | ( | ) |
| : | ; | $ | # | " | ' | { | } | % | . | , |

（2）保留字（Reserved Words）

保留字是用于特殊目的关键字，不区分大小写。保留字在编写程序中要用到，不能作为变量使用。保留字见表 7-3。

表 7-3　S7-SCL 的保留字

| | | |
|---|---|---|
| AND | END_CASE | ORGANIZATION_BLOCK |
| ANY | END_CONST | POINTER |
| ARRAY | END_DATA_BLOCK | PROGRAM |
| AT | END_FOR | REAL |
| BEGIN | END_FUNCTION | REPEAT |
| BLOCK_DB | END_FUNCTION_BLOCK | RETURN |
| BLOCK_FB | END_IF | S5TIME |
| BLOCK_FC | END_LABEL | STRING |
| BLOCK_SDB | END_TYPE | STRUCT |
| BLOCK_SFB | END_ORGANIZATION_BLOCK | THEN |
| BLOCK_SFC | END_REPEAT | TIME |
| BOOL | END_STRUCT | TIMER |
| BY | END_VAR | TIME_OF_DAY |
| BYTE | END_WHILE | TO |
| CASE | ENO | TOD |
| CHAR | EXIT | TRUE |
| CONST | FALSE | TYPE |
| CONTINUE | FOR | VAR |
| COUNTER | FUNCTION | VAR_TEMP |
| DATA_BLOCK | FUNCTION_BLOCK | UNTIL |
| DATE | GOTO | VAR_INPUT |
| DATE_AND_TIME | IF | VAR_IN_OUT |
| DINT | INT | VAR_OUTPUT |
| DIV | LABEL | VOID |
| DO | MOD | WHILE |
| DT | NIL | WORD |
| DWORD | NOT | XOR |
| ELSE | OF | Names of the standard functions |
| ELSIF | OK | |
| EN | OR | |

（3）标识符（Identifiers）

标识符分配给 S7-SCL 语言对象的名称，即是给变量或块等分配的名称。标识符可以最多 24 个字母或者数字组成，其第一个必须是字母或者下划线，不区分大小，但标识符不可为关键字或者标准标识符。

如 X1、_001 以及 Value1 都是合法的标识符，而 001R（第一个字符是数字）、Array

（Array 是关键字）及 X Value（字符间不能有空格）是非法的标识符。由于不区分标识符，Y1 和 y1 是同一个标识符。

S7－SCL 中定义了标准的标识符，分为四大类：块标识符、地址标识符、定时器标识符和计数器标识符。以下详细说明。

1）块标识符（Block Identifier）。块标识符用于块的绝对寻址，与 STEP 7 中的一致，块标识符见表 7–4。

**表 7–4　S7－SCL 的块标识符**

| SIMATIC 标识符 | IEC 标识符 | 含　　义 |
| --- | --- | --- |
| DBx | DBx | 数据块，DB0 为 S7－SCL 保留 |
| FBx | FBx | 功能块 |
| FCx | FCx | 功能 |
| OBx | OBx | 组织块 |
| SDBx | SDBx | 系统数据块 |
| SFCx | SFCx | 系统功能 |
| SFBx | SFBx | 系统功能 |
| Tx | Tx | 定时器 |
| UDTx | UDTx | 自定义数据类型 |
| Zx | Cx | 计数器 |

2）地址标识符（Address Identifier）。在程序的任何位置，都要用地址标识符对 CPU 的存储器进行寻址。地址标识符，如 I0.0、Q0.0 及 M0.1 等与 STEP 7 中定义的存储器一致，请参考前面的章节。

定时器标识符（Timer Identifier）和计数器标识符（Counter Identifier）均与 STEP 7 中基本一致。其表示方法如 T1 和 C1 等。

（4）数字（Numbers）

在 S7－SCL 中，有多种表达数字的方法，其表达规则如下：

1）数字可以有正负、小数点或者指数表达。

2）数字间不能有空格、逗号和空格。

3）为了便于阅读可以用下划线分隔符，如 16#11FF_AAFF 与 16#11FFAAFF 相等。

4）数字前面可以有正号“＋”和负号“－”；没有正负号，默认为正数。

5）数字不可超出范围，如整数范围是－32768～＋32767。

数字中有整数和实数。

整数分为 INT（范围是－32768 ～＋32767）和 DINT（范围是－2147483648 ～＋214748364），合法的整数表达举例：－18、＋188。

实数也称为浮点数，即是带小数点的数，合法的实数表达如 2.3、－1.88 和 1.1e＋3（就是 $1.1\times10^3$）。

（5）字符串（Character Strings）

字符串就是按照一定顺序排列的字符和数字，字符串用单引号标注，如‘QQ&360’。

（6）注释（Comment Section）

注释用于解释程序，帮助读者理解程序，不影响程序的执行，下载程序时，注释不会下

载到 CPU 中去。对程序详细的注释是良好的习惯。

注释从"＊（"开始，到"＊)"结束，注释的例子如下：

```
TEMP1：=1；
(＊这是一个临时变量，
用于存储中间结果＊)
TEMP2 =3；
```

注释还可以用"//"引导，注释长度限制到 254 个字符，但注释前面必须要有"//"。注释的例子如下：

```
VAR_IN_OUT
  TEMP1：=INT；          //I/O Parameters
END_VAR
```

（7）变量（Variables）

在 S7－SCL 中，每个变量在使用前必须声明其变量的类型，根据不同区域将变量分为三类：局域变量、全局变量和允许预定义的变量。

局域变量在逻辑块中（FC、FB 及 OB）中定义，只能在块内有效访问，见表 7-5。

**表 7-5　S7－SCL 的局域变量**

| 序号 | 变量 | 说　明 |
|---|---|---|
| 1 | 静态变量 | 静态变量是变量值在块执行期间和执行后保留在背景数据块中，用于保存功能块值，FB 和 SFB 有，而 FC 和 SFC 无 |
| 2 | 临时变量 | 属于逻辑块，不占用静态内存，其值只在执行期间保留。可以同时作为输入变量和输出变量使用 |
| 3 | 块参数 | 是功能块和功能的形式参数，用于在块被调用时传递实际参数。包括输入参数、输出参数和输入/输出参数等 |

全局变量是指可以在程序中任意位置进行访问的数据或数据域。

**2. S7－SCL 的变量和参数声明**

（1）变量和块参数的声明

变量和块在使用前，必须要进行单独声明，也就是指定块标识符或者变量以及变量的类型。

变量的声明举例如下：

```
VALUE1:REAL;                              //单个变量声明为实数
VALUE2, VALUE3,VALUE4,.... :INT;          //多个变量声明为整数
ARR:ARRAY[1..100, 1..10] OF REAL;         //将变量声明为实数型数组
SET:STRUCT                                //声明结构
      MEASARR:ARRAY[1..20] OF REAL;
      SWITCH:BOOL;
END_STRUCT
```

（2）初始化（Initialization）

静态变量和 FB 的输入及输出参数可以在声明时进行初始化，输入/输出参数为基本数据类型时也可以进行初始化，对于简单的变量用赋值符（：=）即可赋初值。

初始化的应用举例如下：

1）例子 1：

```
VALUE:REAL:=20.25;          //将变量赋初值为实数 20.25
```

2）例子 2：

```
VAR
    //静态变量初始化
    INDEX1:INT:=3;
    //数组初始化
    CONTROLLER1:ARRAY [1..2, 1..2] OF INT:= -54, 736, -83,77;
    CONTROLLER2:ARRAY[1..10] OF REAL:=10(2.5);
    //结构初始化
    GENERATOR:STRUCT
        DAT1:REAL:=100.5;
        A1:INT:=10;
        A2:STRING[6]:=' FACTOR' ;
        A3:ARRAY[1..12] OF REAL:=0.0, 10(100.0), 1.0;
    END_STRUCT;
END_VAR
```

（3）声明变量范围（Declaring Views of Variable Ranges）

要访问具有不同数据类型的变量，可以使用“AT”关键字定义变量中变量的范围和视图。视图只在块内可见，不包括接口。视图的使用和其他变量一样。

举例如下：

```
VAR_INPUT
    Buffer:ARRAY[0..255] OF BYTE;
    Frame1 AT Buffer:UDT100;
    Frame2 AT Buffer:UDT200;
END_VAR
```

**3. S7-SCL 常数的声明**

在 S7-SCL 中使用常量有：位常量、数值常量（整数和实数）、字符常量和时间常量（日期、时间周期）等。

（1）声明常量的符号（Declaring Symbolic Names for Constants）

常量无须声明，但可以在声明区给常量赋符号名，使用“CONST”语句声明常量的符号名。

在简单的表达式中，只有 7 种运算符，即为：**、+、-、*、/、DIV 和 MOD。

举例如下：

```
CONST
      Number:=10;
      TIMEOFDAY1:=TIME#1D_1H_10M_22S_2MS;
      NAME:=' SIEMENS' ;
      NUMBER2:=2*5+10*4;
      NUMBER3:=3+NUMBER2;
END_CONST
```

（2）常量的数据类型定义（Data Types for Constants）

常量的数据类型的定义与算术运算或者逻辑运算的数据类型有关。举例说明如下：

```
Int1:=Int2+12345              //"12345" 给定的数据类型为 INT
INT
Real1:=Real2+12345            //"12345" 给定的数据类型是 Real
REAL
```

常数的数据类型种类和使用示例见表 7-6。

**表 7-6　常量的数据类型定义**

| 序号 | 数据类型 | 类型定义符号 | |
|---|---|---|---|
| 1 | BOOL | BOOL#1<br>Bool#false | bool#0<br>BOOL#TRUE |
| 2 | BYTE | BYTE#0<br>Byte#' å | B#2#101<br>b#16#f |
| 3 | WORD | WORD#32768<br>W#2#1001_ 0100 | word#16#f<br>WORD#8#177777 |
| 4 | DWORD | DWORD#16#f000_ 0000<br>DW#2#1111_ 0000_ 1111_ 0000 | dword#32768<br>DWord#8#37777777777 |
| 5 | INT | INT#16#3f_ ff<br>Int#2#1111_ 0000 | int# -32768 inT#8#77777 |
| 6 | DINT | DINT#16#3fff_ ffff<br>DInt#2#1111_ 0000 | dint# -1000_ 0000<br>dinT#8#17777777777 |
| 7 | REAL | REAL#1<br>real#2e4 | real#1. 5<br>real#3. 1 |
| 8 | CHAR | CHAR#A | CHAR#49 |

**4. 运算符**

一个表达式代表一个值，它可以由单个地址（单个变量）或者几个地址（几个变量）利用运算符结合在一起组成。

运算符有优先级，遵循一般算数运算的规律。S7 -SCL 中的运算符见表 7-7。

**表 7-7　S7 -SCL 的运算符**

| 序号 | 类　别 | 名　称 | 运　算　符 | 优　先　级 |
|---|---|---|---|---|
| 1 | 赋值 | 赋值 | := | 11 |
| 2 | 算术运算 | 幂运算<br>自加（符号）<br>自减（符号）<br>乘<br>除<br>模运算<br>除<br>加、减 | **<br>+<br>-<br>*<br>/<br>MOD<br>DIV<br>+、- | 2<br>3<br>3<br>4<br>4<br>4<br>4<br>5 |
| 3 | 比较运算 | 小于<br>大于<br>小于等于<br>大于等于<br>等于<br>不等于 | <<br>><br><=<br>>=<br>=<br><> | 6<br>6<br>6<br>6<br>7<br>7 |
| 4 | 逻辑运算 | 非<br>与 &<br>异或<br>或 | NOT<br>AND、&<br>XOR<br>OR | 3<br>8<br>9<br>10 |
| 5 | (表达式) | (,) | ( ) | 1 |

**5. 表达式**

表达式是为了计算一个终值所用的公式，它由地址（变量）和运算符组成。表达式的规则如下：

1）两个运算符之间的地址（变量）与优先级高的预算结合。

2）按照运算符优先级进行运算。

3）具有相同的运算级别，从左到右运算。

4）表示前的减号表示该标识符乘以 -1。

5）算数运算不能两个或者两个以上连用。

6）圆括号用于越过优先级。

7）算数运算不能用于连接字符或者逻辑运算。

8）左圆括号与右圆括号的个数应相等。

举例如下：

```
IB10                    //地址
A1 AND (A2)             //逻辑运算表达式
(A3) < (A4)             //比较表达式
3 + 3 * 4/2             //算术运算表达式
```

（1）简单表达式（Simple Expressions）

在 S7 - SCL 中，简单表达式就是简单的加减乘除的算式。举例如下：

```
SIMP_EXPRESSION = A * B + D/C - 3 * VALUE1;
```

（2）算术运算表达式（Arithmetic Expressions）

算术表达式是由算术运算符构成的，允许处理数值数据类型。S7 - SCL 的算术运算符及其地址和结果的数据类型见表 7-8。

**表 7-8　S7 - SCL 的算术运算符及其地址和结果的数据类型**

| 序号 | 运算 | 标识符 | 第一个地址 | 第二个地址 | 结果 | 优先级 |
|---|---|---|---|---|---|---|
| 1 | 幂 | ** | ANY_NUM | ANY_NUM | REAL | 2 |
| 2 | 自加 | ++ | ANY_NUM | - | ANY_NUM | 3 |
| | | | TIME | - | TIME | 3 |
| 3 | 自减 | -- | ANY_NUM | - | ANY_NUM | 3 |
| | | | TIME | - | TIME | 3 |
| 4 | 乘法 | * | ANY_NUM | ANY_NUM | ANY_NUM | 4 |
| | | | TIME | ANY_INT | TIME | 4 |
| 5 | 除法 | / | ANY_NUM | ANY_NUM | ANY_NUM | 4 |
| | | | TIME | ANY_INT | TIME | 4 |
| 6 | 整除 | DIV | ANY_INT | ANY_INT | ANY_INT | 4 |
| | | | TIME | ANY_INT | TIME | 4 |
| 7 | 模运算 | MOD | ANY_INT | ANY_INT | ANY_INT | 4 |
| 8 | 加法 | + | ANY_NUM | ANY_NUM | ANY_NUM | 5 |
| | | | TIME | TIME | TIME | 5 |
| | | | TOD | TIME | TOD | 5 |
| | | | DT | TIME | DT | 5 |

（续）

| 序号 | 运算 | 标识符 | 第一个地址 | 第二个地址 | 结果 | 优先级 |
|---|---|---|---|---|---|---|
| 9 | 减 | - | ANY_NUM | ANY_NUM | ANY_NUM | 5 |
| | | | TIME | TIME | TIME | 5 |
| | | | TOD | TIME | TOD | 5 |
| | | | DATE | DATE | TIME | 5 |
| | | | TOD | TOD | TIME | 5 |
| | | | DT | TIME | DT | 5 |
| | | | DT | DT | TIME | 5 |

注意：以上表格中，ANY_INT 指 INT 和 DINT，而 ANY_NUM 指 INT、DINT 和 Real 的数据类型。

（3）比较运算表达式（Comparison Expressions）

比较运算表达式就是比较两个地址中的数值，结果为布尔数据类型。如果布尔运算的结果为真，则结果为 TRUE；如果布尔运算的结果为假，则结果为 FALSE。比较表达式的规则如下：

1）可以进行比较的数据类型有：INT、DINT、REAL、BOOL、BYTE、WORD、DWORD、CHAR 和 STING。

2）对于 DT、TIME、DATE、TOD 等时间数据类型，只能进行同数据类型的比较。

3）不允许 S5TIME 型的比较，如要进行时间比较，必须使用 IEC 的时间。

4）比较表达式可以与布尔规则相结合，形成语句。例如：Value_A >20 AND Value_B <20）。

（4）逻辑运算表达式（Logical Expressions）

逻辑运算符 AND、&、XOR 和 OR 与逻辑地址（布尔型）或数据类型为 BYTE、WORD 和 DWORD 型的变量结合而构成的逻辑表达式。S7 - SCL 的逻辑运算符及其地址和结果的数据类型见表 7-9。

**表 7-9　S7 - SCL 的逻辑运算符及其地址和结果的数据类型**

| 序号 | 运算 | 标识符 | 第一个地址 | 第二个地址 | 结果 | 优先级 |
|---|---|---|---|---|---|---|
| 1 | 非 | NOT | ANY_BIT | - | ANY_BIT | 3 |
| 2 | 与 | AND | ANY_BIT | ANY_BIT | ANY_BIT | 8 |
| 3 | 异或 | XOR | ANY_BIT | ANY_BIT | ANY_BIT | 9 |
| 4 | 或 | OR | ANY_BIT | ANY_BIT | ANY_BIT | 10 |

**6. 赋值**

通过赋值，一个变量接受另一个变量或者表达式的值。在赋值运算符“：=”左边的是变量，该变量接受右边的地址或者表达式的值。

（1）基本数据类型的赋值（Value Assignments with Variables of an Elementary Data Type）

每个变量、每个地址或者表达式都可以赋值给一个变量或者地址。赋值举例如下：

```
//给变量赋值常数
SWITCH_1:= -17;
SETPOINT_1:=100.1;
```

```
QUERY_1:= TRUE;
TIME_1:= T#1H_20M_10S_30MS;
TIME_2:= T#2D_1H_20M_10S_30MS;
DATE_1:= D#1996 - 01 - 10;
//给变量赋值变量
SETPOINT_1:= SETPOINT_2;
SWITCH_2:= SWITCH_1;
//给变量赋值表达式
SWITCH_2:= SWITCH_1 * 3;
END_FUNCTION_BLOCK
//块的赋值
FUNCTION_BLOCK FB12
VAR
SWITCH_1:INT;
SWITCH_2:INT;
SETPOINT_1:REAL;
SETPOINT_2:REAL;
QUERY_1:BOOL;
TIME_1:S5TIME;
TIME_2:TIME;
DATE_1:DATE;
TIMEOFDAY_1:TIME_OF_DAY;
END_VAR
BEGI
```

（2）结构和 UDT 的赋值（Value Assignments with Variables of the Type STRUCT and UDT）

结构和 UDT 是复杂的数据类型，但很常用。可以对其赋值同样的数据类型变量、同样数据类型的表达式以及同样的结构或者结构内的元素。应用举例如下：

```
BEGIN
    //把一个完整的结构赋值给另一个结构
    MEASVAL:= PROCVAL;
    //结构的一个元素赋值给另一个结构的元素
    MEASVAL. VOLTAGE:= PROCVAL. VOLTAGE;
    //将结构元素赋值给变量
    AUXVAR:= PROCVAL. RESISTANCE;
    //把常数赋值给结构元素
    MEASVAL. RESISTANCE:= 4. 5;
    //把常数赋值给数组元素
    MEASVAL. SIMPLEARR[1,2]:= 4;
END_FUNCTION_BLOCK
```

（3）数组的赋值（Value Assignments with Variables of the Type ARRAY）

数组的赋值类似于结构的赋值，数组元素的赋值和完整数组赋值。数组元素赋值就是对单个数组元素进行赋值，这比较常用。当数组元素的数据类型、数组下标以及数组上标都相同时，一个数组可以赋值给另一个数组，这就是完整数组赋值。应用举例如下：

```
FUNCTION_BLOCK FB3
    VAR
    SETPOINTS:ARRAY [0. . 127] OF INT;
    PROCVALS:ARRAY [0. . 127] OF INT;
    CRTLLR:ARRAY [1. . 3, 1. . 4] OF INT;
    components
    CRTLLR_1:ARRAY [1. . 4] OF INT;
```

```
    END_VAR
    BEGIN
    //把一个数组赋值给另一个数组
    SETPOINTS:= PROCVALS;
    //数组元素赋值
    CRTLLR[2]:= CRTLLR_1;
    //数组元素赋值
    CRTLLR [1,4]:= CRTLLR_1 [4];
END_FUNCTION_BLOCK
```

**7. 控制语句**

S7 – SCL 提供的控制语句可分为三类：选择语句、循环语句和跳转语句。

（1） 选择语句（Selective Statements）

选择语句有 IF 和 CASE，其使用方法和 C 语言等高级计算机语言的用法类似，其功能说明见表 7-10。

**表 7-10　S7 – SCL 的选择语句功能说明**

| 序号 | 语句 | 说　　明 |
|---|---|---|
| 1 | IF | 是二选一的语句，判断条件是“TRUE”或者“FALSE”控制程序进入不同的分支进行执行 |
| 2 | CASE | 是一个多选语句，根据变量值，程序有多个分支 |

1） IF 语句。IF 语句是条件，当条件满足时，按照顺序执行；不满足时，跳出，其应用举例如下：

```
IF I1.1 THEN                    //当 I1.1 =1 时,将 N、SUM 赋值为 0,将 OK 赋值为 FALSE
    N:=0;
    SUM:=0;
    OK:= FALSE;
ELSIF START = TRUE THEN
  N:= N + 1;                    //当 START = TRUE 时,执行 N:= N + 1;
  SUM:= SUM + N;                //当 START = TRUE 时,执行 SUM:= SUM + N;
ELSE
  OK:= FALSE;                   //当 START = FALSE 时,执行 OK:= FALSE;
END_IF;                         //结束 IF 条件语句
```

2） CASE 语句。当需要从问题的多个可能操作中选择其中一个执行时，可以选择嵌套 IF 语句来控制选择执行，但是选择过多会增加程序的复杂性，降低程序的执行效率。这种情况下，使用 CASE 语句就比较合适。其应用举例如下：

```
CASE TW OF
    1:DISPLAY:= OVEN_TEMP;          //当 TW =1 时,执行 DISPLAY:= OVEN_TEMP;
    2:DISPLAY:= MOTOR_SPEED;        //当 TW =2 时,执行 DISPLAY:= MOTOR_SPEED;
    3:DISPLAY:= GROSS_TARE;         //当 TW =3 时,执行 DISPLAY:= GROSS_TAR; QW4:= 16
                                      #0003;
        QW4:= 16#0003;
    4..10:DISPLAY:= INT_TO_DINT (TW);   //当 TW =4..10 时,执行 DISPLAY:= INT_TO_
                                          DINT (TW);
    QW4:= 16#0004;                      //当 TW =4..10 时,执行 QW4:= 16#0004;
    11,13,19:DISPLAY:= 99;
    QW4:= 16#0005;
ELSE:
  DISPLAY:= 0;              //当 TW 不等于以上数值时,执行 DISPLAY:= 0;
```

```
  TW_ERROR:=1;          //当 TW 不等于以上数值时,执行 TW_ERROR:=1;
END_CASE;               //结束 CASE 语句
```

（2） 循环语句（Loops）

S7－SCL 提供的循环语句有三种：FOR 语句、WHILE 语句和 REPEAT 语句。其功能说明见表 7-11。

**表 7-11　S7－SCL 的循环语句功能说明**

| 序　　号 | 语　　句 | 说　　明 |
|---|---|---|
| 1 | FOR | 只要控制变量在指定的范围内，就重复执行语句序列 |
| 2 | WHILE | 只要一个执行条件满足，某一语句就周而复始地执行 |
| 3 | REPEAT | 重复执行某一语句，直到终止该程序的条件满足为止 |

1）FOR 语句。FOR 语句的控制变量必须为 INT 或者 DINT 类型的局部变量。FOR 循环语句定义了：指定的初值和终值，这两个值的类型必须与控制变量的类型一致。其应用举例如下：

```
FUNCTION_BLOCK FOR_EXA
    VAR
        INDEX:INT;
        IDWORD:ARRAY [1..50] OF STRING;
    END_VAR
    BEGIN
    FOR INDEX:=1 TO 50 BY 2 DO      //INDEX 初值为 1,终止为 50,步长为 2
        IF IDWORD [INDEX] ≐ KEY THEN
            EXIT;
        END_IF;
    END_FOR;                        //结束 FOR 语句
END_FUNCTION_BLOCK
```

2）WHILE 语句。WHILE 语句通过执行条件来控制语句的循环执行。执行条件是根据逻辑表达式的规则形成的。其应用举例如下：

```
FUNCTION_BLOCK WHILE_EXA
    VAR
        INDEX:INT;
        IDWORD:ARRAY [1..50] OF STRING;
    END_VAR
    BEGIN
    INDEX:=1;
    WHILE INDEX < =50 AND IDWORD[INDEX] < > KEY DO
        INDEX:=INDEX +2;     //当 INDEX < =50 AND IDWORD[INDEX] < > KEY 时,
                             //执行 INDEX:=INDEX +2;
    END_WHILE;               //终止循环
END_FUNCTION_BLOCK
```

3）REPEAT 语句。在终止条件的满足之前，使用 REPEAT 语句反复执行 REPEAT 语句与 UNTIL 之间的语句。终止的条件是根据逻辑表达式的规则形成的。REPEAT 语句的条件判断在循环体执行之后进行，即终止条件得到满足，循环体仍然至少执行一次。其应用举例如下：

```
FUNCTION_BLOCK REPEAT_EXA
```

```
    VAR
        INDEX:INT;
        IDWORD:ARRAY [1..50] OF STRING;
    END_VAR
    BEGIN
    INDEX:=0;
    REPEAT
    INDEX:= INDEX +2;          //循环执行 INDEX:= INDEX +2;
        UNTIL INDEX >50 OR IDWORD[INDEX] ≐ KEY       //直到 INDEX >50 或 IDWORD
                                                       [INDEX] ≐ KEY
    END_REPEAT;                //终止循环
END_FUNCTION_BLOCK
```

（3）程序跳转语句（Program Jump）

在 S7－SCL 中的跳转语句有四种：CONTINUE 语句、EXIT 语句、GOTO 语句和 RETURN 语句。其功能说明见表 7-12。

**表 7-12　S7－SCL 的程序跳转语句功能说明**

| 序　　号 | 语　　句 | 说　　明 |
|---|---|---|
| 1 | CONTINUE | 用于终止当前循环反复执行 |
| 2 | EXIT | 不管循环终止条件是否满足，在任意点退出循环 |
| 3 | GOTO | 使程序立即跳转到指定的标号处 |
| 4 | RETURN | 使得程序跳出正在执行的块 |

1）CONTINUE 语句的应用举例。用一个例子说明 CONTINUE 语句的应用。

```
INDEX:=0;
WHILE INDEX <=100 DO
    INDEX:= INDEX +1;
    IF ARRAY[INDEX] =INDEX THEN
        CONTINUE;              //当 ARRAY[INDEX] =INDEX 时,退出循环
    END_IF;
    ARRAY[INDEX]:=0;
END_WHILE;
```

2）EXIT 语句的应用举例。用一个例子说明 EXIT 语句的应用。

```
FOR INDEX_1:=1 TO 51 BY 2 DO
    IF IDWORD[INDEX_1] ≐ KEY THEN
        INDEX_2:= INDEX_1;   //当 IDWORD[INDEX_1] ≐ KEY ,执行 INDEX_2:= INDEX_1;
        EXIT;                //当 IDWORD[INDEX_1] ≐ KEY ,执行退出循环
    END_IF;
END_FOR;
```

3）GOTO 语句的应用举例。用一个例子说明 GOTO 语句的应用。

```
IF A >B THEN
        GOTO LAB1;             //当 A >B 跳转到 LAB1
ELSIF A >C THEN
        GOTO LAB2;             //当 A >C 跳转到 LAB2
END_IF;
LAB1:INDEX:=1;
        GOTO LAB3;             //当 INDEX:=1 跳转到 LAB3
LAB2:INDEX:=2;
```

**8. S7－SCL 块**

用户可以在 SCL 中编写块，如 OB、FC、FB 和 UDT 等，也可以调用系统的库内提供各种块，如 SFB、SFC 等。

（1） S7－SCL 块的组成

S7－SCL 块一般由以下部分组成：

1） 块由关键字、块序号或者块符号名开始标识，如组织块的标识为“ORGANIZATION_BLOCK OB20”。对于 FC，也要定义类型，用以确定返回值的数据类型。

2） 由关键字“TITLE =”引导可选块标题。

3） 可选块注释可以一行或者多行，每行以“//”开始。

4） 可选块的属性

5） 可选块的系统属性。

6） 块的声明区。

7） 逻辑块中的语句区或数据块中的赋值区。

8） 以关键字结束块。

以下详细说明。

（2） 块的开始与结束

根据块的类型不同，块的开始和结束标识符有所不同。不同类型块的开始与结束见表 7-13。

**表 7-13　不同类型块的开始与结束**

| 序　号 | 标 识 符 | 块 类 型 | 语　法 |
|---|---|---|---|
| 1 | 功能块 | FB | FUNCTION_BLOCK FBxxx<br>……<br>END_FUNCTION_BLOCK |
| 2 | 功能 | FC | FUNCTION FCxxx：INT<br>……<br>END_FUNCTION |
| 3 | 组织块 | OB | ORGANIZATION_BLOCK OBxxx<br>……<br>END_ORGANIZATION_BLOCK |
| 4 | 数据块 | DB | DATA_BLOCK DBxxx<br>……<br>END_DATA_BLOCK |
| 5 | 自定义数据类型 | UDT | TYPE UDTxxx<br>……<br>END_TYPE |

注：表格中的“xxx”是块的序号。

（3） 块的属性

块的属性包括块的类型、版本、块保护和作者等。块的属性见表 7-14。

表 7-14　表的属性

| 序　　号 | 关键字/属性 | 含　　义 | 举　　例 |
|---|---|---|---|
| 1 | TITLE = xxx' | 块标题 | TITLE = Block Title' |
| 2 | VERSION': xxx' | FC | VERSION': 1.0' |
| 3 | KNOW_HOW_PROTECT | 块保护 | KNOW_HOW_PROTECT |
| 4 | AUTHOR： | 作者名 | AUTHOR：XXH |
| 5 | NAME： | 块名称 | NAME：BMW |
| 6 | FAMILY： | 块系列 | FAMILY：CONTROL |

（4）块的注释

可选块注释可以一行或者多行，每行以“//”开始。

（5）块的系统属性

块的系统属性对整个块都有效，其输入规则是在块的开始语句之后立即定义系统属性。块的系统属性的设置由大括号开始，接着最多 24 个字符的标识符，后接“ = ：”，然后可以打印字符，用单括号括起来，在本行结束用分号，下一行继续输入，在最后一行加上另外一个大括号。应用举例：

```
FUNCTION_BLOCK FB100
{S7_m_c：= true' ;
S7_block1：= "big"；}
```

（6）声明区

声明区是逻辑变量、参数、常数以及标号的声明。变量、参数和常数的声明有规定的规范，其规范和区域见表 7-15。

表 7-15　变量、参数和常数的声明的规范和区域

| 序　　号 | 数　　据 | 语　　法 | FB | FC | OB | DB | UDT |
|---|---|---|---|---|---|---|---|
| 1 | 常数 | CONST<br>....<br>END_CONST | X | X | X | – | – |
| 2 | 标号 | LABEL<br>....<br>END_LABEL | X | X | X | – | – |
| 3 | 临时变量 | VAR_TEMP<br>......<br>END_VAR | X | X | X | – | – |
| 4 | 静态变量 | VAR<br>......<br>END_VAR | X | X | – | X | X |
| 5 | 输入参数 | VAR_INPUT<br>......<br>END_VAR | X | X | – | – | – |
| 6 | 输出参数 | VAR_OUTPUT<br>......<br>END_VAR | X | X | – | – | – |
| 7 | 输入/输出参数 | VAR_IN_OUT<br>......<br>END_VAR | X | X | – | – | – |

注意：表格中“X”表示有效，“-”表示无效。

声明的关键字可以用键盘输入，也可用模板插入，常数的模板插入的方法是：单击菜单中的“Insert”→“Block Template”→“Constant”，在源文件编辑器的工作区就会弹出关键字，如图7-3所示。

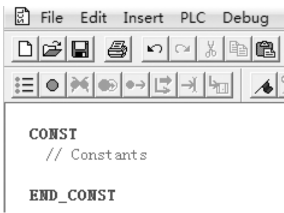



图7-3　常数关键

### 7.1.5　S7-SCL应用举例

在前面的内容中，有较大的篇幅介绍S7-SCL的基础知识，以下通过两个例子介绍S7-SCL的具体应用。第一个例子比较简单。

**【例7-1】** 用S7-SCL语言编写一个功能块FB1，实现将三个整数Add1、Add2和Add3相加，将和存储到Result1中。

**解：**

步骤如下：

1）新建一个项目“SCL1”，并打开SCL源文件编辑器。

2）插入功能块FB1。在SCL源文件编辑器中，单击菜单中的“Insert”→“Block Template”→“FB”，如图7-4所示。再把第一行改为“FUNCTION_BLOCK FB1”。

3）在第一行下面插入注释。这一步可以不做。单击菜单中的“Insert”→“Block Template”→“Comment”，如图7-5所示。图中的圆角方框中的内容已经做了修改，因为是注释，不修改也不会影响结果。

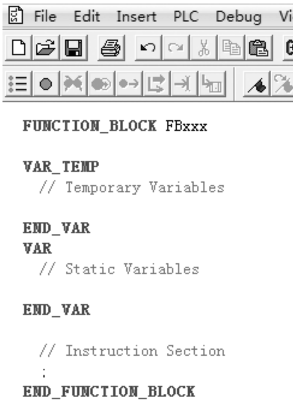



图7-4　插入FB的模板

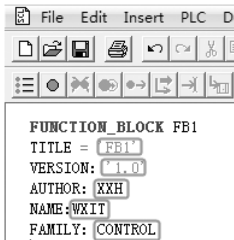



图7-5　插入注释的模板

4）定义输入参数Add1、Add2和Add3的类型为INT，如图7-6所示。

5）定义输出参数Result1的类型为INT，如图7-7所示。

6）编写完整的程序如图7-8所示。

```
VAR_INPUT
  // 输入参数
  Add1:INT;
  Add2:INT;
  Add3:INT;
END_VAR
```

图7-6　定义输入参数

```
VAR_OUTPUT
 Result1:INT;// 输出参数
END_VAR
```

图7-7　定义输出参数

7）编译程序。单击工具栏中的“Compile”按钮，开始编译程序，如果程序无语法错误，则在如图 7-9 所示的左下侧显示“0Errors，0Warning（s）”（0 个错误，0 个警告）。

```
FUNCTION_BLOCK FB1
TITLE = 'FB1'
VERSION: '1.0'
AUTHOR: XXH
NAME: WXIT
FAMILY:CONTROL
VAR_INPUT
  // 输入参数
  Add1:INT;
  Add2:INT;
  Add3:INT;
END_VAR

VAR_OUTPUT
 Result1:INT;// 输出参数
END_VAR

  BEGIN;
  Result1 :=Add1+Add2+Add3;
END_FUNCTION_BLOCK
```

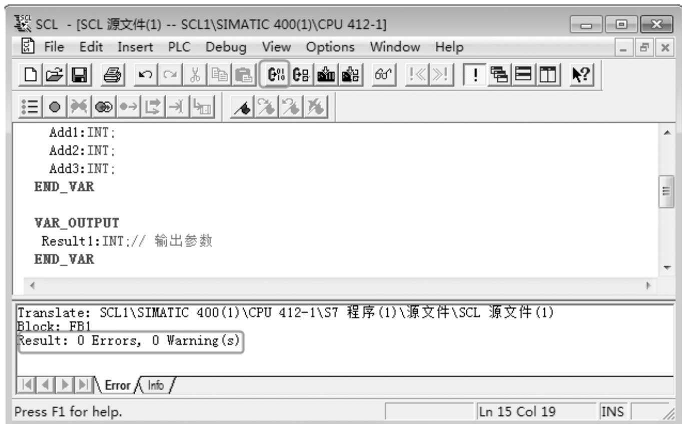



图 7-8　完整的块程序　　　　　　　　　　　　图 7-9　编译

8）在主程序中调用 FB1，如图 7-10 所示。

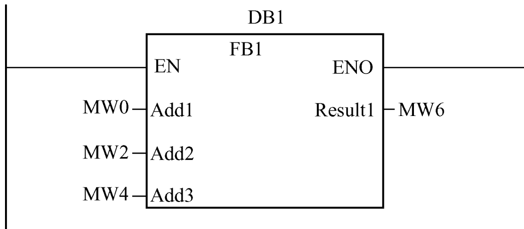



图 7-10　OB1 中的梯形图

9）在 S7－SCL 中，对块 FB1 加密。只要在 SCL 程序中加一个块保护关键字“KNOW_HOW_PROTECT”即可，位置如图 7-11 所示。块 FB1 加密后带锁形标识，如图 7-12 所示。再在图 7-12 中，选中“SCL 源文件（1）”，并将其删除，如图 7-13 所示。“加锁”后的 FB1，不能打开，即为加密，这样处理有利于保护知识产权。

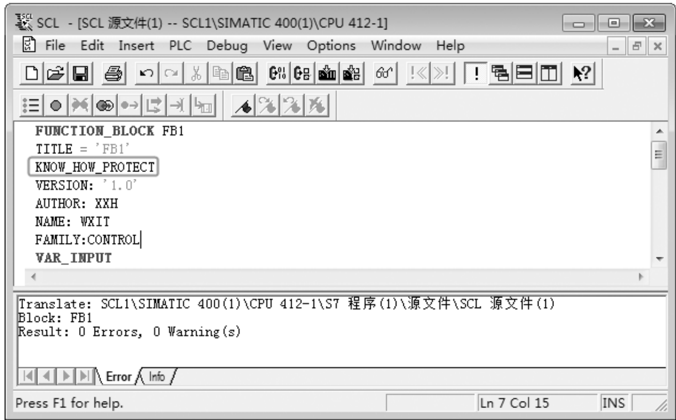



图 7-11　S7－SCL 中对块 FB1 加密




图 7-12　块 FB1 加密后带锁形标识

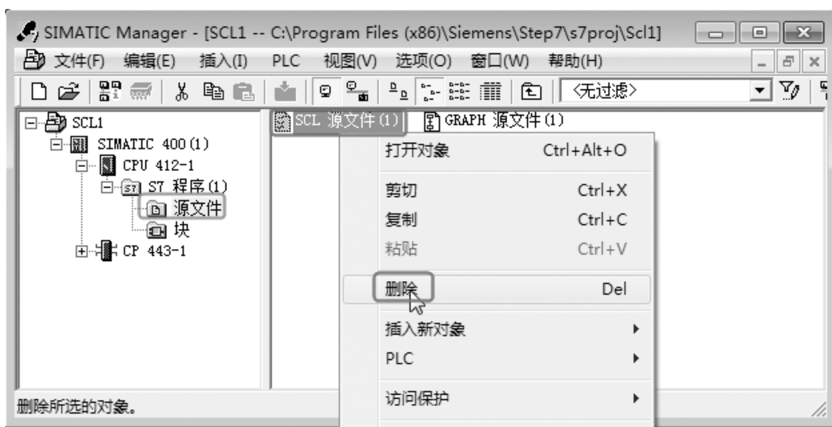



图 7-13　删除 S7 - SCL 源文件

**【例 7-2】**有一个控制系统，要求采集一路温度信号，温度信号的范围 0 ~ 100℃。对于高于 100℃的温度，视作干扰信号，仍然按照 100℃采集；对于低于 0℃的温度，视作干扰信号，仍然按照 0℃采集。要求显示实时温度和历史最高温度两个温度数值。请用 SCL 编写块实现。

**解：**

步骤如下：

1）新建一个项目“SCL3”，并打开 SCL 源文件编辑器。

2）插入模板 FC。在 SCL 源文件编辑器中，单击菜单中的“Insert”→“Block Template”→“FC”。再把第一行改为“FUNCTION FC110：INT1”。

3）插入功能 FC105。单击菜单中的“Insert”→“Block Call”，在标准库中选中功能“FC105”。

4）插入注释、插入参数，然后定义临时变量、输入变量和输出变量。

5）编写程序，如图 7-14 所示。

6）编写主程序，如图 7-15 所示。FC110 从引脚上看，与标准库中的 FC105 很类似，但 FC110 中多了 LValue（最大值参数），而且采集的温度变量范围在 0 ~ 100℃内。

```
FUNCTION FC110: INT
TITLE = 'FC110'
KNOW_HOW_PROTECT
VERSION: '1.0'
AUTHOR: XXH
NAME: WXIT
FAMILY:CONTROL

VAR_TEMP
  // 临时变量
    RET_VAL1:WORD;
END_VAR

 VAR_INPUT
  // 输入参数
  PIW_IN:INT;
  HI_LIM1:REAL:=100.0;
  LO_LIM1:REAL:=0.0;
  BIPOLAR1:BOOL:=FALSE;
END_VAR

VAR_OUTPUT
 // 输出参数
 OUT1:REAL;
 LValue:REAL;
END_VAR

 BEGIN;
    RET_VAL1:=SCALE(IN:=PIW_IN  // IN: INT
      ,HI_LIM:=HI_LIM1 // IN: REAL
      ,LO_LIM:=LO_LIM1  // IN: REAL
      ,BIPOLAR:=BIPOLAR1  // IN: BOOL
      ,OUT:=OUT1  // OUT: REAL
      ); // WORD
IF  PIW_IN >27648 THEN  //高于上限取上限值
      OUT1 :=HI_LIM1;
END_IF ;
IF  PIW_IN <0 THEN    //低于下限时，取下限值
      OUT1  :=LO_LIM1;
END_IF ;
IF OUT1>LValue THEN   //求最大值
    LValue:=OUT1;
END_IF ;
   FC110:= 0;
END_FUNCTION
```

图 7-14　完整的块程序

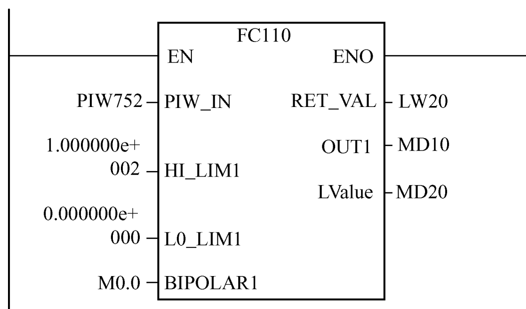



图 7-15　OB1 中的程序

## 7.2　西门子 PLC 的 GRAPH 编程

实际的工业生产的控制过程中，顺序逻辑控制占有相当大的比例。所谓顺序逻辑控制，就是按照生产工艺预先规定的顺序，在各个输入信号的作用下，根据内部状态和时间顺序，在生产过程中的各个执行机构自动地、有秩序地进行操作。S7－GRAPH 是一种顺序功能图编程语言，它能有效地应用于设计顺序逻辑控制程序。

### 7.2.1　S7－GRAPH 简介

S7－GRAPH 是一种顺序功能图编程语言，适合用于顺序逻辑控制。S7－GRAPH 有如下特点：

① 适用于顺序控制程序。

② 符合国际标准 IEC 61131－3。

③ 通过了 PLCopen 基础级认证。

④ 适用于 SIMATIC S7－300（推荐用于 CPU314 以上 CPU）、S7－400、C7 和 WinAC。

S7－GRAPH 针对顺序控程序做了相应优化处理，它不仅仅具 PLC 典型的元素（例如：输入/输出、定时器及计数器），而且增加了如下一些概念：

多个顺控器（最多 8 个）；步骤（每个顺控器最多 250 个）；每个步骤的动作（每步最多 100 个）；转换条件（每个顺控器最多 250 个）；分支条件（每个顺控器最多 250 个）；逻辑互锁（最多 32 个条件）；监控条件（最多 32 个条件）；事件触发功能；切换运行模式（手动、自动及点动模式）。

## 7.2.2　S7－GRAPH 的安装与兼容性

**1. S7－GRAPH 的安装**

STEP 7 标准版并不包括 S7－GRAPH 软件包及授权，需单独购买，STEP 7 Professional 版包括了 S7－GRAPH 的软件包及授权，安装即可。在 S7 程序中，S7－GRAPH 块可以与其他 STEP 7 编程语言生成的块组合互相调用，S7－GRAPH 生成的块也可以作为库文件被其他语言引用。

**2. S7－GRAPH 与操作系统的兼容性**

S7－GRAPH 是否能安装到操作系统中，与 S7－GRAPH 版本和操作系统版本密切相关，这就是兼容性。S7－GRAPH 与操作系统的兼容性见表 7-16。

**表 7-16　S7－GRAPH 与操作系统的兼容性**

| 产 品 版 本 | STEP 7 V5. 5 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | WinXP SP2 | WinXP SP3 | Win7 | Win7 SP1 | Win7 (64) | Win7 sp1 (64) | Win2008 R2 (64) | Win2008 R2 SP1 (64) |
| S7－GRAPH V5. 3 | X | – | – | – | – | – | – | – |
| S7－GRAPH V5. 3 SP6 | X | X | X | – | – | – | – | – |
| S7－GRAPH V5. 3 SP7 | X | X | X | X | X | X | X | X |

注：1. 表中的“X”表示兼容。
2. “－”表示不兼容。
3. 举例：Win7（64）操作系统需要安装 S7－GRAPH V5. 3 SP7 以上版本的软件包。

## 7.2.3　S7－GRAPH 的应用基础

**1. S7 程序构成**

作为 STEP 7 的选项包，S7－GRAPH 软件在安装后，将被集成在 STEP 7 中使用。S7－GRAPH 编程界面为图形界面，包含若干个顺控器。当编译 S7－GRAPH 程序时，其生成的块以 FB 的形式出现，此 FB 可以被其他程序调用，例如 OB1、OB35。顺序控制 S7 程序构成如图 7-16 所示。

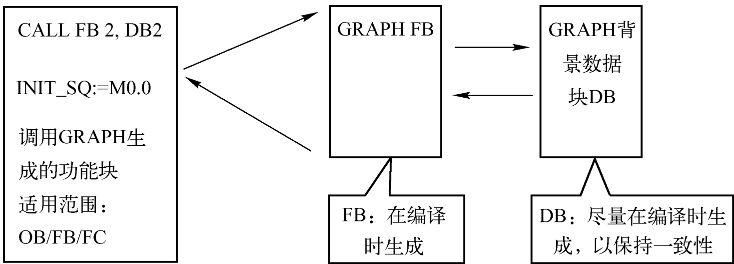



图 7-16　顺序控制 S7 程序构成

**2. S7 – GRAPH 的编辑器**

在 SIMATIC Manager 界面中，选中“块”，然后执行菜单命令，单击“插入”→“功能块”，打开功能块 FB1 的属性对话框，如图 7–17 所示。




图 7–17　FB1 的属性对话框

在名称中输入“FB1”，在创建语言中输入“GRAPH”，单击“OK”按钮。FB1 已经创建，双击“FB1”，打开 GRAPH 编辑器，如图 7–18 所示。

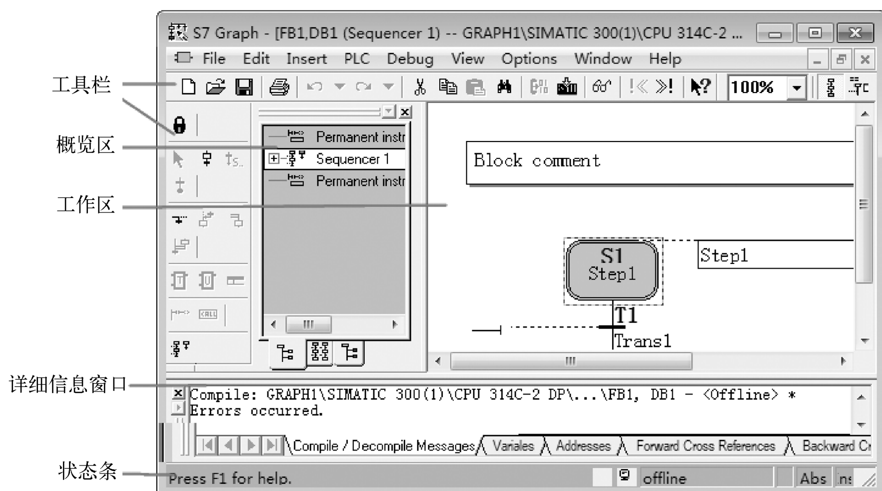



图 7–18　GRAPH 编辑器

S7 – GRAPH 编辑器由生成和编辑程序的工作区、工具栏（标准工具栏、浮动工具栏和视窗工具栏）、详细信息窗口、概览区和状态条等组成。

（1） 工具条

工具条中可以分为 4 类功能，具体如下：

① 标准功能：主要为文件功能，例如打开、保存、复制和粘贴等功能，与 Office 中的此类工具类似。

② 视图功能 ：调整显示作用，如是否显示符号名等。

③ 顺控器：包含顺控器元素，如分支、跳转和步等。

④ LAD/FBD：可以为每步添加 LAD/FBD 指令。

（2）工作区

工作区是 S7 - GRAPH 中最重要的区域，它是顺序功能图编辑区。

（3）概览窗口

1）概览窗口的 Graphic 视图。如图 7-19 所示，Permanent instructions 的英文原意为永久的指令。可以通过鼠标右键单击图中的区域，在提示“Insert New Element”→“Permanent Instruction”下，选择“Condition”或者“Call”。不论顺控器的状态如何，这些指令可以在每次扫描时被执行。可以在 S7 - GRAPH 的 FB 中编写任意多的指令。

Permanent instructions 只支持两种格式，即：

- LAD/FBD 格式的 AND、OR 和比较指令（每个 NETWORK 最多可以容纳 32 个）。
- CALL 指令。

如果希望 Permanent instructions 使用 LAD/FBD 其他格式指令，将这些指令编写为 FB、FC，然后用 CALL 格式调用。

2）概览窗口的 Sequencer 视图。对于每个 S7 - GRAPH 的 FB 程序，可以包括多个 Sequencer（顺控器），如果希望对多个顺控器概览查看，可以选择概览窗口的 Sequencer 视图，如图 7-20 所示。

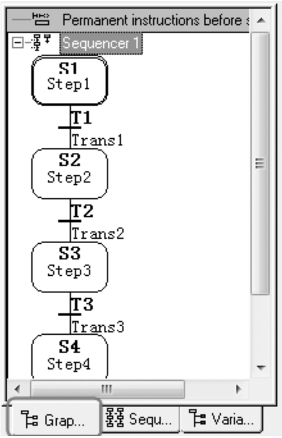



图 7-19　概览窗口的 Graphic 视图

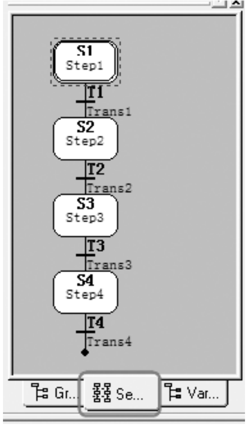



图 7-20　概览窗口的 Sequencer 视图

3）概览窗口的 Variables 视图。在 S7 - GRAPH 的 FB 程序中，可以通过概览窗口的 Variables 视图，查看并编辑如图 7-21 所示的资源。

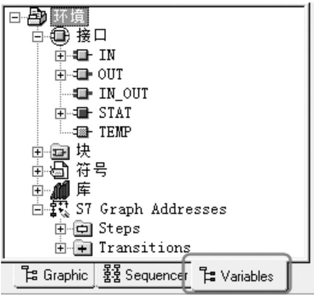



图 7-21　概览窗口的 Variables 视图

4）详细信息窗口。可以查看编译信息、变量监控、交叉参考等信息。

5）状态条。可以显示在线（online）、离线（off-line）和 CPU 状态等信息。

**3. 顺控器规则**

S7 - GRAPH 格式的 FB 程序是这样工作的：

- 每个 S7 - GRAPH 格式的 FB，都可以作为一个普通 FB 被其他程序调用。
- 每个 S7 - GRAPH 格式的 FB，都被分配一个背景数据块，此数据块用来存储 FB 参数设置及当前状态等。
- 每个 S7 - GRAPH 格式的 FB，都包括三个主要部分：顺控器之前的 Permanent instructions、一个或多个顺控器以及顺控器之后的 Permanent instructions。

（1）顺控器执行规则

1）步的开始。每个顺控器都以一个初始步或者多个位于顺控器任意位置的初始步开始。

只要某个步的某个动作（action）被执行，则认为此步被激活（active）；如果多个步被同时执行，则认为是多个步被激活（active）。

2）一个激活的步的退出。任意激活的干扰（active disturbs），例如互锁条件或监控条件的消除或确认，并且至后续步的转换条件（transition）满足时，激活步退出。

3）满足转换条件的后续步被激活。

4）在顺控器的结束位置的处理。

- 如有一个跳转指令（jump），指向本顺控器的任意步，或者 FB 的其他顺控器。此指令可以实现顺控器的循环操作。
- 如有分支停止指令，顺控器的步将停止。

5）激活的步（Active Step）。激活的步是一个当前自身的动作正在被执行的步。一个步在如下任意情况下，都可被激活：

- 当某步前面的转换条件满足。
- 当某步被定义为初始步（initial step），并且顺控器被初始化。
- 当某步被其他基于事件的动作调用（event - dependent action）。

（2）顺控器的结构

顺控器主要结构有：简单的线性结构顺控器（如图 7-22a 所示）、选择结构及并行结构顺控器（如图 7-22b 所示）和多个顺控器（如图 7-22c 所示）。

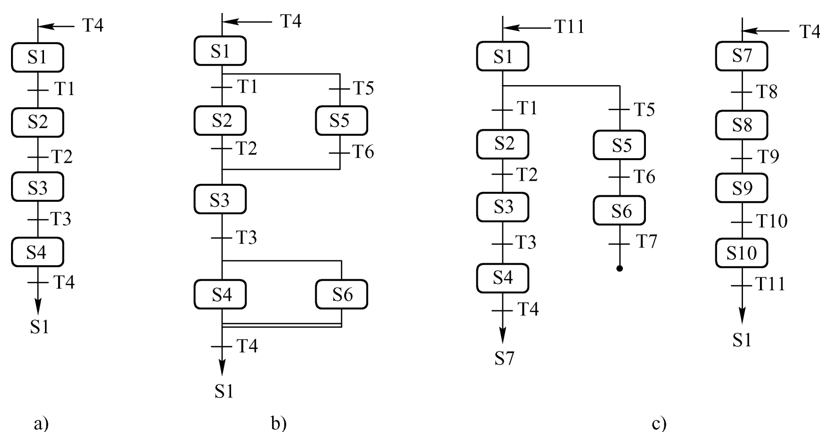



图 7-22　顺控器的结构

（3）顺控器元素

在工具栏中有一些顺控器元素是创建程序所必需的。顺控器元素详细介绍见表 7-17。

**表 7-17　顺控器元素的含义**

| 序　　号 | 元　　素 | 英 文 含 义 | 中 文 含 义 |
|---|---|---|---|
| 1 | | Step + Transition | 步 + 转换条件 |
| 2 | | Jump | 跳转指令 |
| 3 | | Open Alternative Branch | 打开选择分支 |
| 4 | | Close Alternative Branch | 关闭选择分支 |
| 5 | | Open Simultaneous Branch | 打开并行分支 |
| 6 | | Close Simultaneous Branch | 关闭并行分支 |
| 7 | | Branch Stop | 分支停止 |
| 8 | | Insert Sequencer | 插入顺控器 |

**4. 条件与动作的编程**

（1）步的构成及属性。

一个 S7 - GRAPH 的程序由多个步组成，其中每一步由步序、步名、转换编号、转换名、转换条件以及动作命令组成，如图 7-23 所示。步序、步名和转换名由系统自动生成，一般无须修改，也可以自己修改，但必须是唯一的。转换条件可以使 LAD 或者 FBD 指令编程，可以在 S7 - GRAPH 编辑器的菜单“View”中切换，例如单击“View”→“LAD”，就是 LAD 编程。步的动作由命令和操作数地址组成，左边的框中输入命令，右边的框中输入操作数地址。

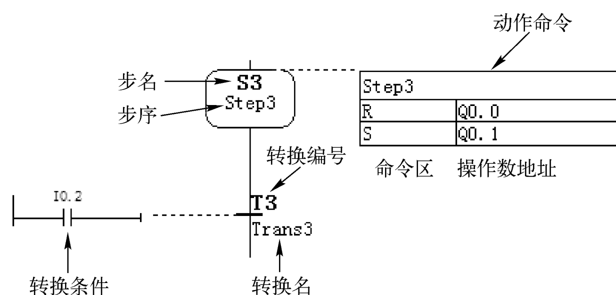


图 7-23　步的说明图

（2）动作（action）

动作由标准动作和事件有关的动作，动作中可以为定时器、计数器和算数运算等。步的动作在 S7 - GRAPH 的 FB 中占有重要位置，用户大部分控制任务要由步的动作来完成，编程者应当熟练掌握所有的动作指令。添加动作很容易，选中动作框，单击鼠标右键，再单击快捷菜单中的“Insert New Element”→“Action”，即可添加动作，添加动作如图 7-24 所示。

标准动作在编写程序中较为常用，常用的标准动作含义见表 7-18。

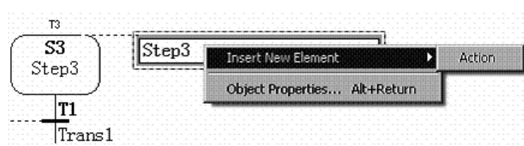



图 7-24 添加动作

**表 7-18 常用的标准动作的含义**

| 序　号 | 命　令 | 含　义 |
|---|---|---|
| 1 | N | 输出。当该步为激活步时，对应的操作数输出为 1；当该步为非激活步时，对应的操作数输出为 0 |
| 2 | S | 置位。当该步为激活步时，对应的操作数输出为 1；当该步为非激活步时，对应的操作数输出为 1，除非遇到某一激活步将其复位 |
| 3 | R | 复位。当该步为激活步时，对应的操作数输出为 0，并一致保持 |
| 4 | D | 延迟。当该步为激活步时，开始倒计时，计时时间到，对应的操作数输出为 1；当该步为非激活步时，对应的操作数输出为 0 |
| 5 | L | 脉冲限制。当该步为激活步时，对应的操作数输出为 1，并开始倒计时，计时时间到，输出为 0；当该步为非激活步时，对应的操作数输出为 0 |
| 6 | CALL | 块调用。当该步为激活步时，指定的块会被调用 |

（3）动作中的定时器

时间出现时，定时器将被执行，联锁功能也能用于定时器。

TL 为扩展脉冲定时器命令，该命令的下面一行是定时器时间，定时器没有闭锁功能。

TD 命令用于实现由闭锁功能的延迟。一旦事件发生，定时器被起动。联锁条件 C 仅仅在定时器起动的时刻起作用。

（4）动作中的计数器

动作中的计数器的执行与指定的计数事件有关，对于有联锁功能的计数器，只有联锁条件满足和指定的事件出现时，动作中的计数器才会计数。计数器命令和联锁组合时，命令后面要加“C”。

事件发生时，计数器指令 CS 将初值装入计数器。CS 的下面一行是要装入的计数器的初始值。事件发生时，CU、CD 和 CR 指令，分别使得计数器加 1、减 1 和复位。

（5）动作中的算数运算

在动作中可以使用如下简单的算数运算语句：

① A：=B。

② A：=函数（B），可以使用 S7-GRAPH 内置的函数。

③ A：=B <运算符> C，例如 A：=B+C。

算数运算必须使用英文符号，不允许使用中文符号。

**5. 转换条件**

转换条件可以是事件（例如退出激活步），也可以是状态变化。条件可以在转换、联锁、监控和永久性指令中出现。

**6. S7-GRAPH 的功能块参数设置**

在 S7-GRAPH 编辑器中，单击“Option”→“Block Setting”，可打开 S7-GRAPH 的

功能块参数设置对话框，如图 7-25 所示。

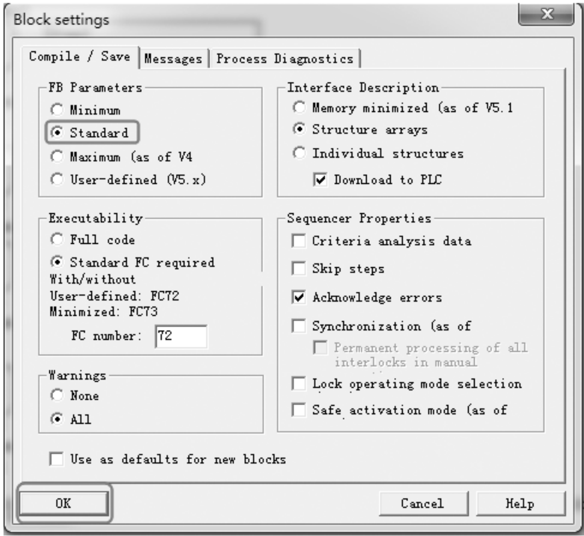



图 7-25　S7 - GRAPH 编辑器功能块参数设置

在 FB Parameters 区域有 4 个参数设置区有 4 个参数集选项，分别介绍如下：

1）Minimum（最小参数集）。FB 只包括 SQ_INIT 起动参数，如图 7-26a 所示，如果用户的程序仅仅会运行在自动模式，并且不需要其他的控制及监控功能。

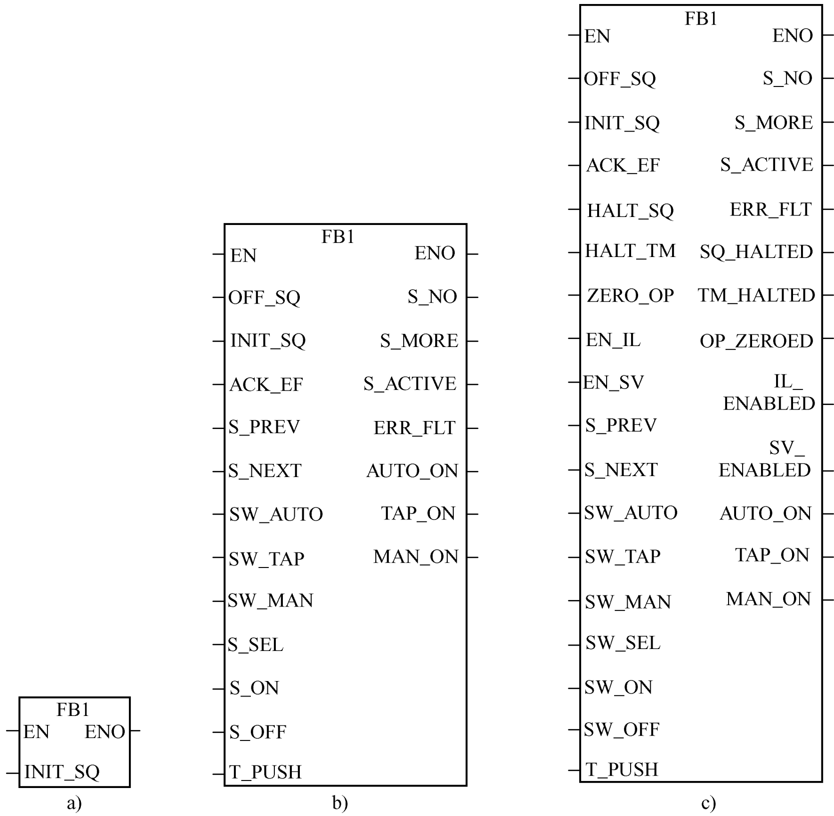



图 7-26　不同参数集对应的功能块形式

2）Standard（标准参数集）。FB 包括默认参数，如图 7-26b 所示，如果用户希望程序运行在各种模式，并提供反馈及确认消息功能。

3） Maximum（最大参数集）。FB 包括默认参数、扩展参数、提供更多的控制和监控参数，如图 7-26c 所示。

4） User - defined（用户定义参数集）。包括默认参数和扩展参数，可提供更多的控制和监控参数。

S7 - GRAPH FB 的部分参数及其含义见表 7-19。

**表 7-19　S7 - GRAPH FB 的参数及其含义**

| 序　号 | FB 参数 | 数 据 类 型 | 含　义 |
|---|---|---|---|
| 1 | ACK_EF | BOOL | 故障信息得到确认 |
| 2 | INIT_SQ | BOOL | 激活初始步，顺控器复位 |
| 3 | OFF_SQ | BOOL | 停止顺控器，例如使所有步失效 |
| 4 | SW_AUTO | BOOL | 模式选择：自动模式 |
| 5 | SW_MAN | BOOL | 模式选择：手动模式 |
| 6 | SW_TAP | BOOL | 模式选择：单步调节 |
| 7 | SW_TOP | BOOL | 模式选择：自动或切换到下一个 |
| 8 | S_SEL | INT | 选择用于输出参数 S_ON 指定的步，在 S_ON 中显示手动模式在 S_ON 参数中指定的下一步 |
| 9 | S_ON | BOOL | 手动模式：激活步显示 |
| 10 | S_OFF | BOOL | 手动模式：使显示步变成激活步 |
| 11 | T_PUSH | BOOL | 单步调节模式：如果条件满足时，实现转换 |
| 12 | ERROR | BOOL | 错误显示：“互锁” |
| 13 | FAULT | BOOL | 错误显示：“监视” |
| 14 | EN_SSKIP | BOOL | 激活步的跳转 |
| 15 | EN_ACKREQ | BOOL | 使能确认需求 |
| 16 | SQ_HALT | BOOL | 暂停顺序控制器 |
| 17 | TM_HALT | BOOL | 停止所有步的激活运行时间和块运 |
| 18 | ZERO_OP | BOOL | 复位所有在激活步 N、D、L 操作到 0 |
| 19 | EN_IL | BOOL | 复位/重新使能步互锁 |
| 20 | EN_SV | BOOL | 复位/重新使能步监视 |

### 7.2.4　S7 - GRAPH 的应用举例

以下通过简单的例子来讲解 S7 - GRAPH 编程应用的全过程。

**【例 7-3】**用一台 PLC 控制 4 盏灯，实现如下功能：

初始状态时所有的灯都不亮；按下按钮 SB1，灯 HL1 亮；按下 SB2 按钮，灯 HL2 亮，HL1 灭；按下 SB3 按钮，灯 HL3 亮，HL2 灭；2 s 后，灯 HL3 和灯 HL4 亮；再 2 s 后，灯 HL3 和灯 HL4 熄灭，灯 HL1 亮；如此循环。

程序要求用 S7－GRAPH 语言编写功能块实现。

**解：**

解题步骤如下：

1）根据题意，先绘制流程图如图 7-27 所示。

2）新建一个项目“GRAPH1”，并进行硬件组态，最后编译和保存。

3）插入功能块 FB1。在 SIMATIC Manager 界面中，选中“块”，然后执行菜单命令，单击“插入”→“功能块”，打开功能块 FB1 的属性对话框，如图 7-28 所示。

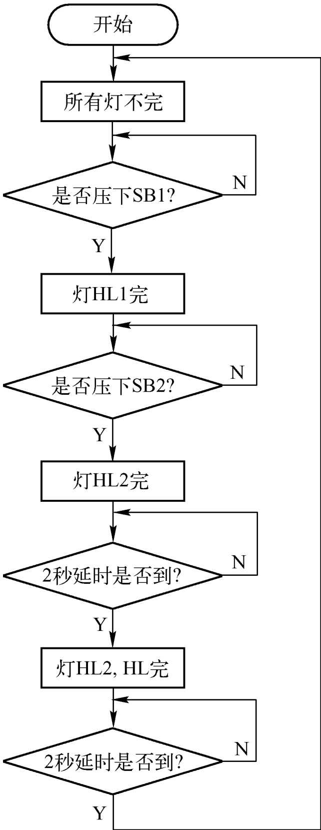



图 7-27　流程图




图 7-28　FB1 的属性对话框

4）编辑 GRAPH 程序。在名称中输入“FB1”，在创建语言中输入“GRAPH”，单击“OK”按钮。FB1 已经创建，双击“FB1”，打开 GRAPH 编辑器，弹出编辑界面，选中“T1”处，鼠标右键单击快捷菜单的“Insert New Element”→“Step + Transition”，插入“步＋转换条件”，如图 7-29 所示。

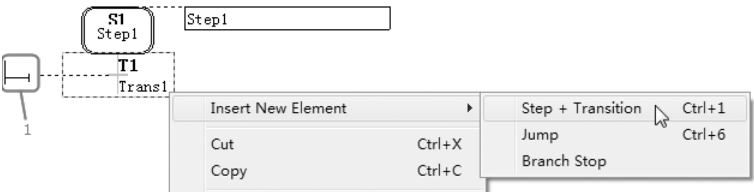



图 7-29　插入“步＋转换条件”

选中如图 7-29 中的标记“1”处，在单击左侧的工具栏的“常开触点”按钮╢├，并在“常开触点”上面输入 I0.0，如图 7-30 所示。

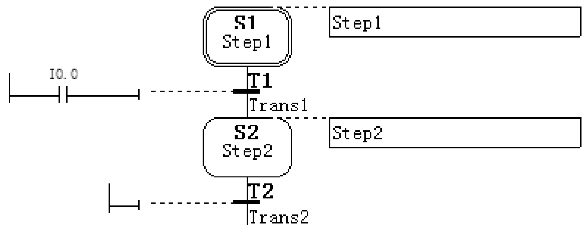



图 7-30　插入“常开触点”

如图 7-31 所示的“Step2”，鼠标右键单击快捷菜单的“Insert New Element”→“Action”，插入“动作”。

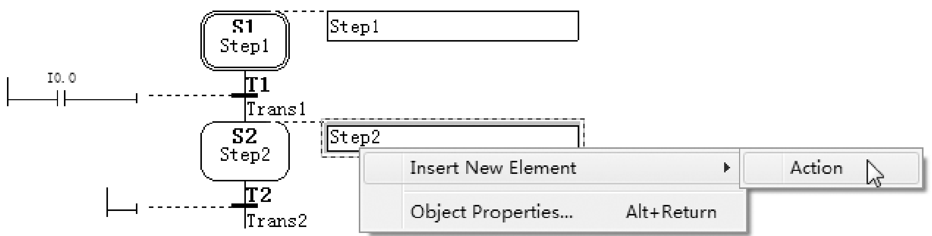



图 7-31　插入“动作”

在动作命令框的左侧输入命令 N，右侧输入操作数 Q0.0，如图 7-32 所示。

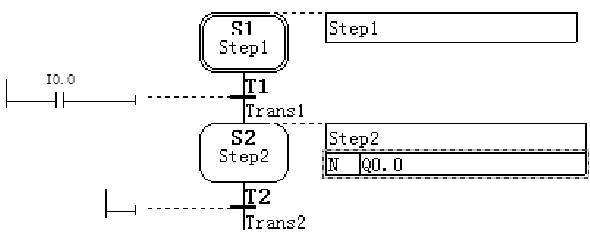



图 7-32　输入“动作”命令和操作数

编写完整的 GRAPH 程序如图 7-33 所示。之后，单击标准工具栏中的“保存”按钮，这个步骤非常重要，在 SIMATIC Manager 界面，如图 7-34，可以看到“块”中有“FC72”和“SFC64”两个块，这是单击“保存”按钮后系统自动生成的。

5）编写 OB1 中的程序。将功能块 FB1 拖入程序编辑区，编写程序如图 7-35 所示。

6）运行仿真。把程序下载到仿真器 S7-PLCSIM 中，将“I0.0”置为“1”。再切换到 S7-GRAPH 编辑器界面，单击标准工具栏中的“Monitor”按钮，处于监控状态下，FB1 中的 GRAPH 程序如图 7-36 所示。

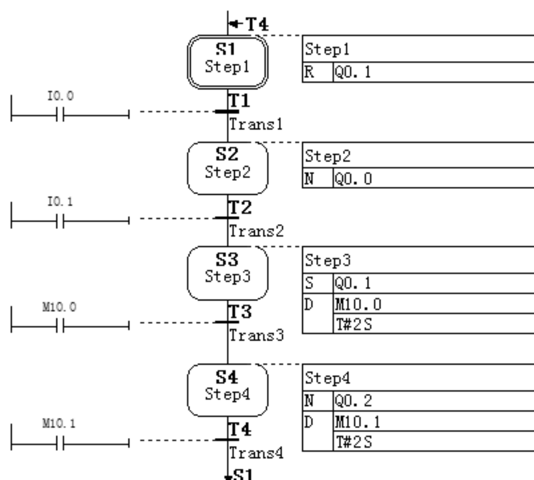



图 7-33　FB1 中完整的 GRAPH 程序




图 7-34　SIMATIC Manager 界面

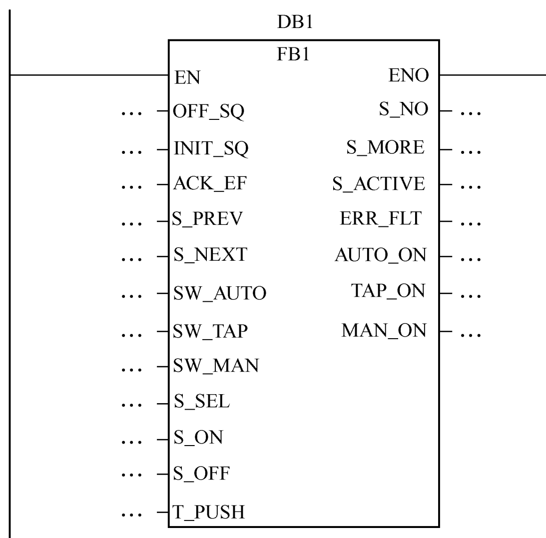



图 7-35　OB1 中的程序

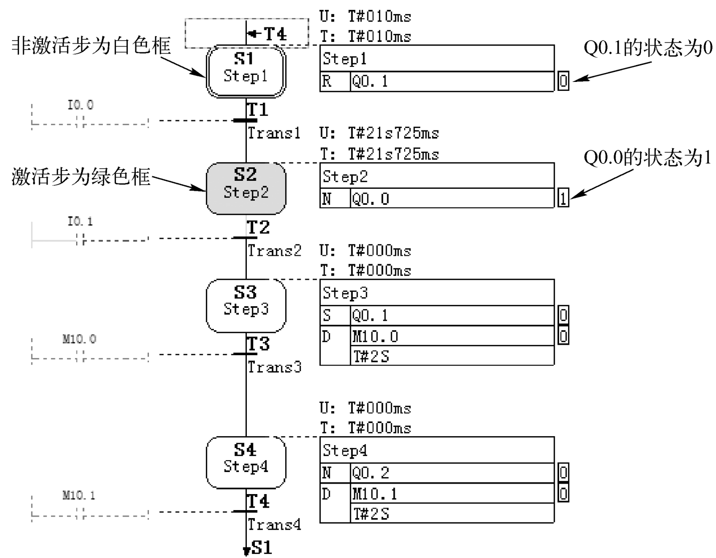



图 7-36　FB1 中的 GRAPH 程序

# 第 8 章　S7－300/400 PLC 的通信及其应用

本章介绍 S7－300/400 PLC 的通信基础知识，并用实例介绍 S7－200 与 S7－300、S7－300 与 S7－300、S7－300/400 与 S7－400 PLC 之间的 MPI 通信；S7－300 与 ET200M、S7－200、S7－300 与 S7－300/400 PLC 之间的 PROFIBUS 通信；S7－300 与 ET200M、S7－300/400 PLC 之间的以太网通信。

## 8.1　通信基础知识

PLC 的通信包括 PLC 与 PLC 之间的通信、PLC 与上位计算机之间的通信以及和其他智能设备之间的通信。PLC 与 PLC 之间通信的实质就是计算机的通信，使得众多独立的控制任务构成一个控制工程整体，形成模块控制体系。PLC 与计算机连接组成网络，将 PLC 用于控制工业现场，计算机用于编程、显示和管理等任务，构成“集中管理、分散控制”的分布式控制系统（DCS）。

### 8.1.1　通信的基本概念

**1. 串行通信与并行通信**

串行通信和并行通信是两种不同的数据传输方式。

串行通信就是通过一对导线将发送方与接收方进行连接，传输数据的每个二进制位，按照规定顺序在同一导线上依次发送与接收，如图 8-1 所示。例如，常用的优盘 USB 接口就是串行通信。串行通信的特点是通信控制复杂，通信电缆少，因此与并行通信相比成本低。

并行通信就是将一个 8 位数据（或 16 位、32 位）的每一个二进制位采用单独的导线进行传输，并将传送方和接收方进行并行连接，一个数据的各二进制位可以在同一时间内一次传送，如图 8-2 所示。例如，老式打印机的打印口和计算机的通信就是并行通信。并行通信的特点是一个周期里可以一次传输多位数据，其连线的电缆多，因此长距离传送时成本高。

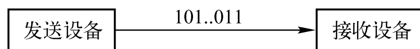



图 8-1　串行通信

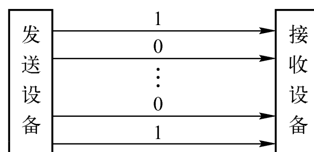



图 8-2　并行通信

**2. 异步通信与同步通信**

异步通信与同步通信也称为异步传送与同步传送，这是串行通信的两种基本信息传送方式。从用户的角度上说，两者最主要的区别在于通信方式的“帧”不同。

异步通信方式又称起止方式。它在发送字符时，要先发送起始位，然后是字符本身，最后是停止位，字符之后还可以加入奇偶校验位。异步通信方式具有硬件简单、成本低的特

点，主要用于传输速率低于 19.2 Kbit/s 以下的数据通信。

同步通信方式在传递数据的同时，也传输时钟同步信号，并始终按照给定的时刻采集数据。其传输数据的效率高，硬件复杂，成本高，一般用于传输速率高于 20 Kbit/s 以上的数据通信。

**3. 单工、全双工与半双工**

单工、双工与半双工是通信中描述数据传送方向的专用术语。

1）单工（Simplex）：指数据只能实现单向传送的通信方式，一般用于数据的输出，不可以进行数据交换，如图 8-3 所示。

2）全双工（Full Simplex）：也称双工，指数据可以进行双向数据传送，同一时刻既能发送数据，也能接收数据，如图 8-4 所示。通常需要两对双绞线连接，通信线路成本高。例如，RS－422 就是“全双工”通信方式。



图 8-3　单工通信

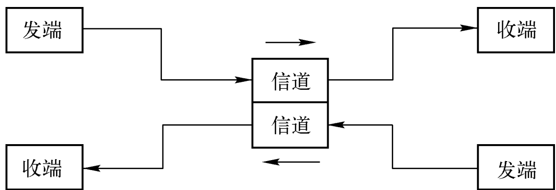


图 8-4　双工通信

3）半双工（Half Simplex）：指数据可以进行双向数据传送，同一时刻，只能发送数据或者接收数据，如图 8-5 所示。通常需要一对双绞线连接，与全双工相比，通信线路成本低。例如，RS－485 只用一对双绞线时就是“半双工”通信方式。

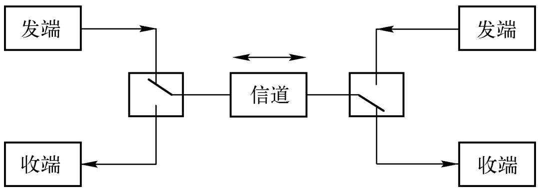


图 8-5　半双工通信

## 8.1.2　PLC 网络的术语解释

PLC 网络中的名词、术语很多，现将常用的予以介绍。

1）站（Station）：在 PLC 网络系统中，将可以进行数据通信、连接外部输入/输出的物理设备称为“站”。例如，由 PLC 组成的网络系统中，每台 PLC 可以是一个“站”。

2）主站（Master Station）：PLC 网络系统中进行数据连接的系统控制站，主站上设置了控制整个网络的参数，通常每个网络系统只有一个主站，站号实际就是 PLC 在网络中的地址。

3）从站（Slave Station）：PLC 网络系统中，除主站外，其他的站称为“从站”。

4）远程设备站（Remote Device Station）：PLC 网络系统中，能同时处理二进制位、字

的从站。

5）本地站（Local Station）：PLC 网络系统中，带有 CPU 模块并可以与主站以及其他本地站进行循环传输的站。

6）站数（Number of Station）：PLC 网络系统中，所有物理设备（站）所占用的“内存站数”的总和。

7）网关（Gateway）：又称网间连接器、协议转换器。网关在传输层上以实现网络互联，是最复杂的网络互联设备，仅用于两个高层协议不同的网络互联。如图 8-6 所示，S7－400 通过工业以太网，把信息传送到 IE/PB 模块，再传送到 PROFIBUS 网络上的 IM153－1 模块，IE/PB 通信模块用于不同协议的互联，它实际上就是网关。

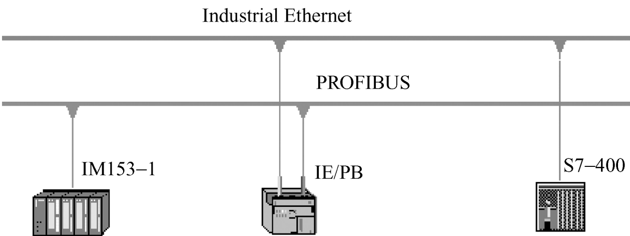



图 8-6　网关应用实例

8）中继器（Repeater）：用于网络信号放大、调整的网络互联设备，能有效延长网络的连接长度。例如，PPI 的正常传送距离是不大于 50 m，经过中继器放大后，可传输 1 千多米，应用实例如图 8-7 所示，PLC 通过 MPI 或者 PPI 通信时，传送距离可达 1100 m。




图 8-7　中继器应用实例

9）路由器（Router，转发者）：所谓路由就是指通过相互连接的网络把信息从源地点移动到目标地点的活动。一般来说，在路由过程中，信息至少会经过一个或多个中间节点。路由器是互联网的主要节点设备。如图 8-8 所示，如果要把 PG/PC 的程序下载到 SIMATIC 400(1)中，必然要经过 SIMATIC 300(1)这个节点，这实际就用到了 SIMATIC 3004 的路由功能，图中的箭头展示了下载信息的流向。

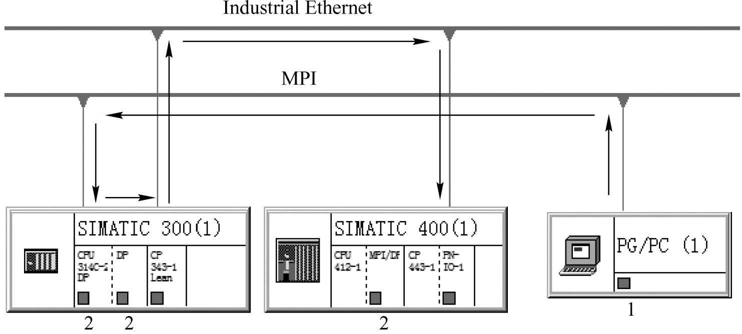



图 8-8　路由功能应用实例

10）交换机（Switch）：交换机是为了解决通信阻塞而设计的，它是一种基于 MAC 地址识别、能完成封装转发数据包功能的网络设备。交换机可以“学习”MAC 地址，并把其存放在内部地址表中，通过在数据帧的始发者和目标接收者之间建立临时的交换路径，使数据帧直接由源地址到达目的地址。如图 8-9 所示，交换机（ESM）将 HMI（触摸屏）、PLC 和 PC（个人计算机）连接在工业以太网网的一个网段中。

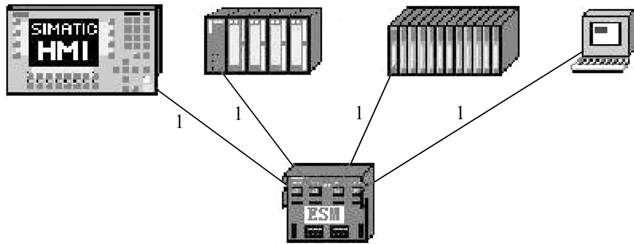



图 8-9　交换机应用实例

11）网桥（Bridge）：也叫桥接器，是连接两个局域网的一种存储/转发设备，它能将一个大的 LAN 分割为多个网段，或将两个以上的 LAN 互联为一个逻辑 LAN，使 LAN 上的所有用户都可访问服务器。网桥将网络的多个网段在数据链路层连接起来，网桥的应用如图 8-10 所示。西门子的 DP/PA Coupler 模块就是一种网桥。

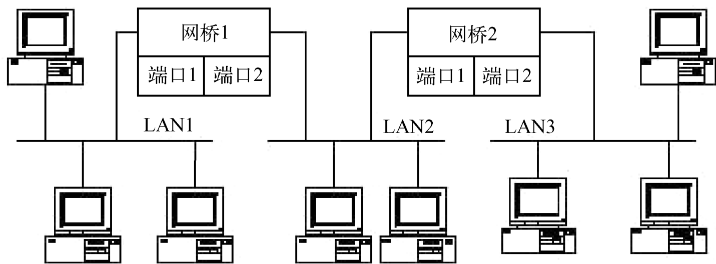



图 8-10　网桥应用实例

### 8.1.3　RS－485 标准串行接口

**1. RS－485 接口**

RS－485 接口是在 RS－422 基础上发展起来的一种 EIA 标准串行接口，采用“平衡差分驱动”方式。RS－485 接口满足 RS－422 的全部技术规范，可以用于 RS－422 通信。RS－485 接口通常采用 9 针连接器，其外观与引脚定义如图 8-11 所示。RS－485 接口的引脚功能参见表 8-1。

**表 8-1　RS－485 接口的引脚功能**

| PLC 侧引脚 | 信号代号 | 信号功能 |
|---|---|---|
| 1 | SG 或 GND | 机壳接地 |
| 2 | +24 V 返回 | 逻辑地 |
| 3 | RXD＋或 TXD＋ | RS－485 的 B，数据发送/接收＋端 |
| 4 | 请求－发送 | RTS(TTL) |

（续）

| PLC 侧引脚 | 信 号 代 号 | 信 号 功 能 |
|---|---|---|
| 5 | +5 V 返回 | 逻辑地 |
| 6 | +5 V | +5 V |
| 7 | +24 V | +24 V |
| 8 | RXD - 或 TXD - | RS - 485 的 A，数据发送/接收 - 端 |
| 9 | 不适用 | 10 位协议选择（输入） |

**2. 西门子的 PLC 连线**

西门子 PLC 的 PPI 通信、MPI 通信和 PROFIBUS - DP 现场总线通信的物理层都是 RS - 485，而且采用都是相同的通信线缆和专用网络接头。图 8-12 显示了电缆接头的终端状况，右端的电阻设置为“on”，而左侧的设置为“off”，图中只显示了一个，若有多个也是这样设置。要将终端电阻设置“on”或者“off”，只要拨动网络接头上的拨钮即可。图 8-12 中拨钮在“on”一侧，因此终端电阻已经接入电路。

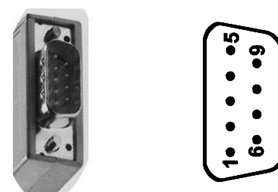


图 8-11　网络接头的外观与引脚定义

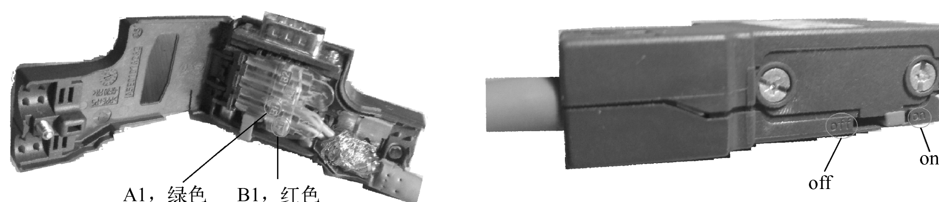


图 8-12　网络接头的终端电阻设置图

**【关键点】** 西门子的专用 PROFIBUS 电缆中有两根线，一根为红色，上标有“B”，一根为绿色，上面标有“A”，这两根线只要与网络接头上相对应的“A”和“B”接线端子相连即可（如“A”线与“A”接线端相连）。网络接头直接插在 PLC 的通信口上即可，不需要其他设备。注意：三菱的 FX 系列 PLC 的 RS - 485 通信要加 RS - 485 专用通信模块和终端电阻。

## 8.1.4　OSI 参考模型

通信网络的核心是 OSI（Open System Interconnection，开放式系统互联）参考模型。1984 年，国际标准化组织（ISO），提出了开放式系统互联的 7 层模型，即 OSI 模型。该模型自下而上分为：物理层、数据链接层、网络层、传输层、会话层、表示层和应用层。

OSI 的上三层通常称为应用层，用来处理用户接口、数据格式和应用程序的访问。下四层负责定义数据的物理传输介质和网络设备。OSI 参考模型定义了大多数协议栈共有的基本框架，如图 8-13 所示。

1）物理层（Physical Layer）：定义了传输介质、连接器和信号发生器的类型，规定了物理连接的电气、机械功能特性，如电压、传输速率、传输距离等特性。建立、维护、断开物

理连接。典型的物理层设备有集线器（HUB）和中继器等。

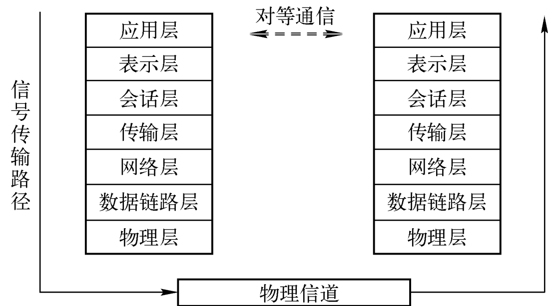



图 8-13　信息在 OSI 模型中的流动形式

2）数据链路层（Data Link Layer）：确定传输站点物理地址以及将消息传送到协议栈，提供顺序控制和数据流向控制。建立逻辑连接、进行硬件地址寻址、差错效验等功能（由底层网络定义协议）。典型的数据链路层的设备有交换机和网桥等。

3）网络层（Network Layer）：进行逻辑地址寻址，实现不同网络之间的路径选择。协议有：ICMP IGMP IP（IPV4 IPV6）、ARP 及 RARP。典型的网络层设备是路由器。

4）传输层（Transport Layer）：定义传输数据的协议端口号，以及流控和差错效验。协议有：TCP、UDP。网关是互联网设备中最复杂的，它是传输层及以上层的设备。

5）会话层（Session Layer）：建立、管理及终止会话。

6）表示层（Presentation Layer）：数据的表示、安全及压缩。

7）应用层（Application）：网络服务与最终用户的一个接口。协议有：HTTP、FTP、TFTP SMTP、SNMP 及 DNS。

数据经过封装后通过物理介质传输到网络上，接收设备除去附加信息后，将数据上传到上层堆栈层。

## 8.1.5　SIMATIC NET 工业网络

SIMATIC NET 是西门子工业通信网络解决方案的总称。SIMATIC 中的网络如图 8-14 所示。

西门子通信网络技术说明如下：

（1）MPI 通信

MPI（Multi-Point Interface，即多点接口）协议，用于小范围、少点数的现场级通信。MPI 是为 S7/M7/C7 系统提供接口，它设计用于编程设备的接口，也可用于在少数 CPU 间传递少量的数据。MPI 通信的应用在趋于减少。

（2）PROFIBUS 通信

PROFIBUS 符合国际标准 IEC61158，是目前国际上通用的现场总线中 20 大现场总线之一，并以其独特的技术特点、严格的认证规范、开放的标准和众多的厂家支持等优势，成为现场级通信网络的优秀解决方案，目前其全球网络节点已经突破 3000 万个。

从用户的角度看，PROFIBUS 提供三种通信协议类型：PROFIBUS-FMS、PROFIBUS-DP 和 PROFIBUS-PA。

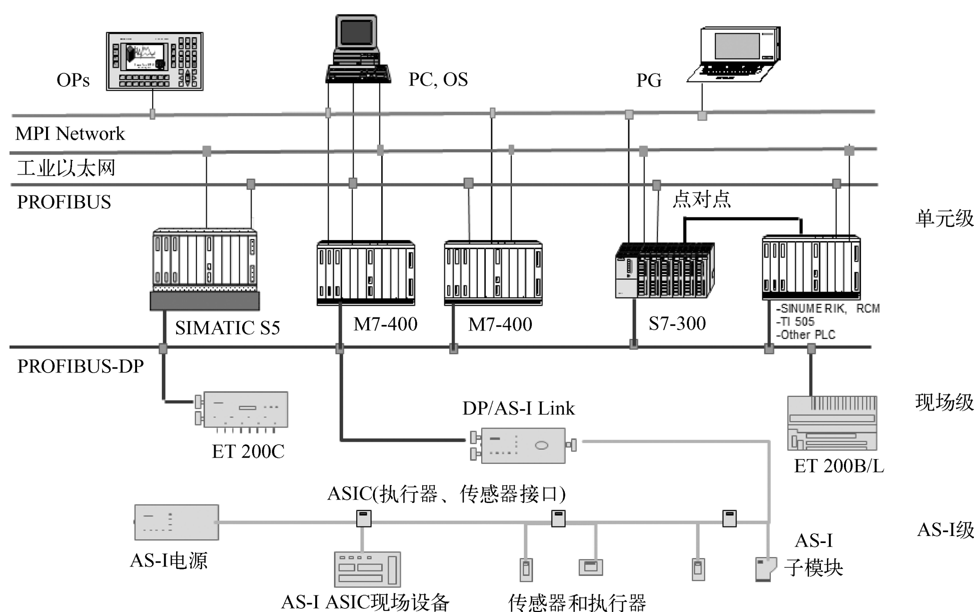



图 8-14 SIMATIC 中的网络

（3）工业以太网

工业以太网符合 IEEE802.3 国际标准，是功能强大的区域和单元网络，也是目前工控界最为流行的网络通信技术之一。

（4）点对点连接

严格地说，点对点（Point－to－Point）连接并不是网络通信。但点对点连接可以通过串口连接模块实现数据交换，应用比较广泛。

（5）AS－Interface

传感器/执行器接口用于自动化系统最底层的通信网络。它专门用来连接二进制的传感器和执行器，每个从站的最大数据量为 4 bit。

## 8.2 现场总线概述

### 8.2.1 现场总线的概念

**1. 现场总线的诞生**

现场总线是 20 世纪 80 年代中后期在工业控制中逐步发展起来的。计算机技术的发展为现场总线的诞生奠定了技术基础。

另一方面，智能仪表也出现在工业控制中。智能仪表的出现为现场总线的诞生奠定了应用基础。

**2. 现场总线的概念**

国际电工委员会（IEC）对现场总线（Fieldbus）的定义为：一种应用于生产现场，在现场设备之间、现场设备和控制装置之间实行双向、串行、多节点的数字通信网络。

现场总线的概念有广义与狭义之分。狭义的现场总线是指基于 EIA485 的串行通信网络。广义的现场总线泛指用于工业现场的所有控制网络。广义的现场总线包括狭义现场总线和工业以太网。

### 8.2.2 主流现场总线的简介

1984 年国际电工技术委员会/国际标准协会（IEC/ISA）就开始制定现场总线的标准，然而统一的标准至今仍未完成。很多公司推出其各自的现场总线技术，但彼此的开放性和互操作性难以统一。

经过 12 年的讨论，终于在 1999 年年底通过了 IEC61158 现场总线标准，这个标准容纳了 8 种互不兼容的总线协议。后来又经过不断讨论和协商，在 2003 年 4 月，IEC61158 Ed.3 现场总线标准第 3 版正式成为国际标准，确定了 10 种不同类型的现场总线为 IEC61158 现场总线。2007 年 7 月，第四版现场总线增加到 20 种。见表 8-2。

**表 8-2　IEC61158 的现场总线**

| 类型编号 | 名　称 | 发起的公司 |
|---|---|---|
| Type 1 | TS61158 现场总线 | 原来的技术报告 |
| Type 2 | ControlNet 和 Ethernet/IP 现场总线 | 美国 Rockwell 公司 |
| Type 3 | PROFIBUS 现场总线 | 德国 Siemens 公司 |
| Type 4 | P-NET 现场总线 | 丹麦 Process Data 公司 |
| Type 5 | FF HSE 现场总线 | 美国 Fisher Rosemount 公司 |
| Type 6 | SwiftNet 现场总线 | 美国波音公司 |
| Type 7 | World FIP 现场总线 | 法国 Alstom 公司 |
| Type 8 | INTERBUS 现场总线 | 德国 Phoenix Contact 公司 |
| Type 9 | FF H1 现场总线 | 现场总线基金会 |
| Type 10 | PROFINET 现场总线 | 德国 Siemens 公司 |
| Type 11 | TC net 实时以太网 | |
| Type 12 | Ether CAT 实时以太网 | 德国倍福 |
| Type 13 | Ethernet Powerlink 实时以太网 | 最大的贡献来自于 Alstom |
| Type 14 | EPA 实时以太网 | 中国浙大、沈阳所等 |
| Type 15 | Modbus RTPS 实时以太网 | 施耐德 |
| Type 16 | SERCOS Ⅰ、Ⅱ 现场总线 | 数字伺服和传动系统数据通信 |
| Type 17 | VNET/IP 实时以太网 | 法国 Alstom 公司 |
| Type 18 | CC-Llink 现场总线 | 三菱电机公司 |
| Type 19 | SERCOS Ⅲ 现场总线 | 数字伺服和传动系统数据通信 |
| Type 20 | HART 现场总线 | Rosemount 公司 |

### 8.2.3 现场总线的特点

现场总线系统具有以下特点：

① 系统具有开放性和互用性。
② 系统功能自治性。
③ 系统具有分散性。
④ 系统具有对环境的适应性。

### 8.2.4 现场总线的现状

现场总线的现状有如下几点：
① 多种现场总线并存。
② 各种总线都有其应用的领域。
③ 每种现场总线都有其国际组织和支持背景。
④ 多种总线已成为国家和地区标准。
⑤ 一个设备制造商通常参与多个总线组织。
⑥ 各个总线彼此协调共存。

### 8.2.5 现场总线的发展

现场总线技术是控制、计算机和通信技术的交叉与集成，几乎涵盖了连续和离散工业领域，如过程自动化、制造加工自动化、楼宇自动化以及家庭自动化等。它的出现和快速发展体现了控制领域对降低成本、提高可靠性、增强可维护性和提高数据采集智能化的要求。现场总线技术的发展趋势体现在四个方面：
① 统一的技术规范与组态技术是现场总线技术发展的一个长远目标。
② 现场总线系统的技术水平将不断提高。
③ 现场总线的应用将越来越广泛。
④ 工业以太网技术将逐步成为现场总线技术的主流。

## 8.3 MPI 通信及其应用

### 8.3.1 MPI 通信简介

多点接口（Multi Point Interface，MPI）网络可用于单元层，是西门子公司开发的用于 PLC 之间通信的保密的协议。MPI 通信是当通信速率要求不高、通信数据量不大时，可以采用的一种简单经济的通信方式。

MPI 主要的优点是 CPU 可以同时与多个设备建立通信联系。也就是说，编程器、HMI 设备和其他的 PLC 可以连接在一起并同时运行。编程器通过 MPI 接口生成的网络还可以访问所连接硬件站上的所有智能模块。可同时连接的其他通信对象的数目取决于 CPU 的型号。例如，CPU314 的最大连接数为 4，CPU416 为 64。

MPI 接口的主要特性如下：

- RS－485 物理接口。
- 传输率为 19.2 Kbit/s 或 187.5 Kbit/s 或 1.5 Mbit/s。
- 最大连接距离为 50 m（2 个相邻节点之间），有两个中继器时为 1100 m，如图 8-15 所

示。采用光纤和星形偶合器时为 23.8 km。

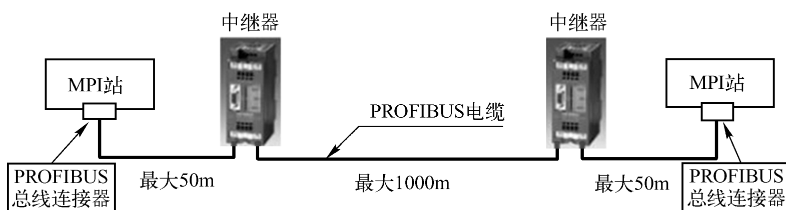



图 8-15　MPI 通信（带中继器）

● 采用 PROFIBUS 元件（电缆、连接器）。

MPI 通信有全局数据通信、基本通信和扩展通信，以下将分别介绍：

① 全局数据通信。这种通信方法通过 MPI 接口在 CPU 间循环地交换数据，而不需要编程。当过程映像被刷新时，在循环扫描检测点上进行数据交换。对于 S7-400，数据交换可以用 SFC 来起动。全局数据可以是输入、输出、标志位、定时器、计数器和数据块区。

数据通信不需要编程，而是利用全局数据表来配置。不需要 CPU 的连接用于全局数据通信。

② 基本通信。这种通信方法可用于所有 S7-300/400，它通过 MPI 子网或站中的 K 总线来传送数据。系统功能（SFC），例如 X_SEND（在发送端）和 X_RCV（在接收端）被用户程序调用。最大用户数据量为 76B。当系统功能被调用时，通信连接被动态地建立和断开。在 CPU 上需要有一个自由的连接。

③ 扩展通信。这种通信方法可用于所有的 S7-400 PLC。通过任何子网（MPI、PROFIBUS、Industrial Ethernet）可以传送最多 64 KB 的数据。它是通过系统功能块（SFB）来实现的，支持有应答的通信。数据也可以读出或写入到 S7-300（PUT/GET 块）。不仅可以传送数据，而且可以执行控制功能，例如控制通信对象的起动和停机。这种通信方法需要配置连接（连接表）。该连接在一个站的全起动时建立并且一直保持。在 CPU 上需要有自由的连接。

### 8.3.2　S7-200 与 S7-300 间的 MPI 通信

S7-200 与 S7-300 间的 MPI 通信只能采用单边无组态通信，也就是通信无需组态。以下用一个例子说明这种通信的方法。

**【例 8-1】** 有两台设备，分别由一台 CPU 314C-2DP 和一台 CPU 226CN 控制，从设备 1 上的 CPU 314C-2DP 发出起停控制命令，设备 2 的 CPU 226CN 收到命令后，对设备 2 进行起停控制。同理设备 2，也能发出信号，对设备 1 进行起停控制。

**解：** 将设备 1 上的 CPU314C-2DP 作为主站，主站的 MPI 地址为 2，将设备 2 上的 CPU 226CN 作为从站，从站的 MPI 地址为 3。

**1. 主要软硬件配置**

① 1 套 STEP 7 V5.5 SP4。

② 1 台 CPU 314C-2DP。

③ 1 台 CPU 226CN。

④ 1 台 EM277。

⑤ 1 根编程电缆。

⑥ 1 根 PROFIBUS 网络电缆（含两个网络总线连接器）。

⑦ 1 套 STEP 7 Micro/WIN V4.0 SP9。

MPI 通信硬件配置图如图 8-16 所示，PLC 接线图如图 8-17 所示。

从图 8-17 可以看出 S7－200 与 S7－300 间的 MPI 通信有两种配置方案。方案 1 只要将 PROFIBUS 网络电缆（含两个网络总线连接器）连接在 S7－300 的 MPI 接口和 S7－200 的 PPI 接口上即可，而方案 2 却需要另加一个 EM277 模块，显然成本多一些，但若 S7－200 的 PPI 接口不够用时，方案 2 是可以选择的配置方案。

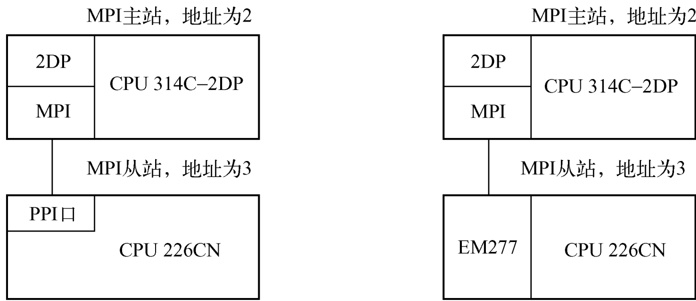



图 8-16　MPI 通信硬件配置图

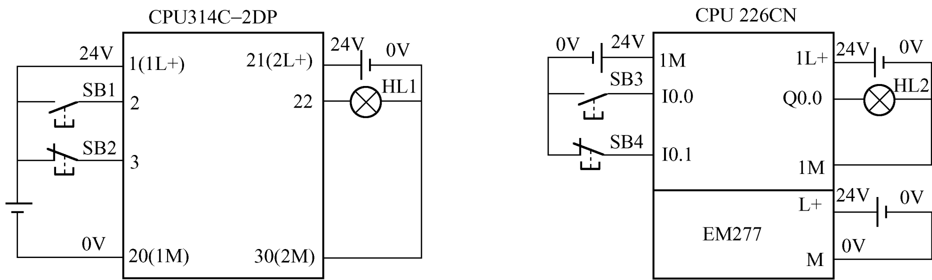



图 8-17　PLC 接线图

**2. 硬件组态**

S7－200 与 S7－300 间的 MPI 通信只能采用无组态通信，无组态通信指通信无须组态，完成通信任务仅需编写程序即可。只要用到 S7－300，硬件组态还是不可缺少的，这点读者必须清楚。

1）新建项目并插入站点。新建项目，命名为“MPI_200”，再插入站点，重命名为“MASTER”，如图 8-18 所示，双击“硬件”，打开硬件组态界面。




图 8-18　新建项目并插入站点

2）组态主站硬件。先插入导轨，再插入 CPU 模块，如图 8-19 所示，双击“CPU 314C-2DP”，打开 MPI 通信参数设置界面，单击“属性”按钮，如图 8-20 所示。

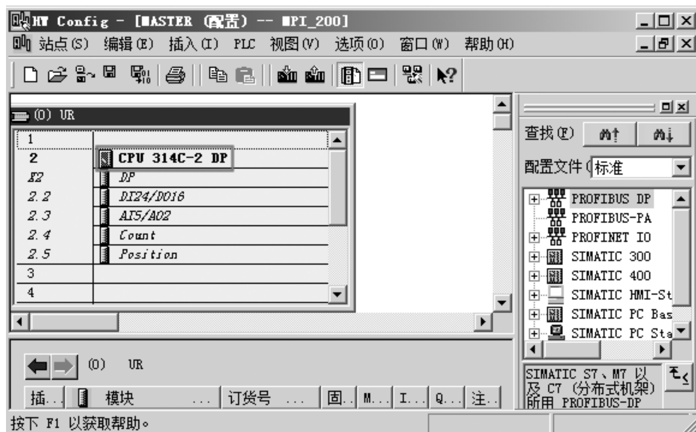



图 8-19　组态主站硬件

3）设置主站的 MPI 通信参数。先选定 MPI 的通信波特率为默认的“187.5 Kbps”，再选定主站的 MPI 地址为“2”，然后单击“确定”按钮，如图 8-21 所示。最后编译保存和下载硬件组态，在此不再重复叙述。

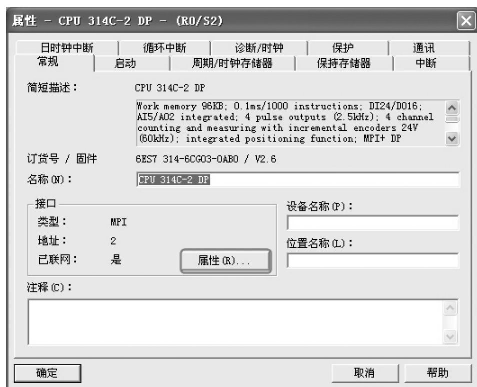



图 8-20　打开 MPI 通信参数设置界面

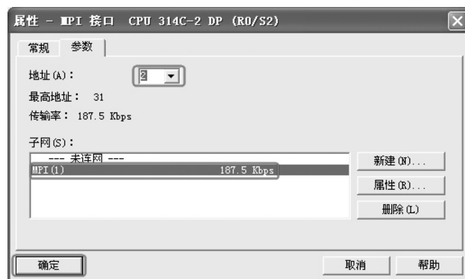



图 8-21　设置主站的 MPI 通信参数

4）在 SIMATIC 管理器界面中，插入组织块 OB100 和 OB35，如图 8-22 所示。




图 8-22　插入组织块 OB100 和 OB35

5）打开系统块。完成以上步骤后，S7－300 的硬件组态完成，但还必须设置 S7－200 的通信参数。先打开 STEP 7－Micro/WIN 软件，选定工具条中的“系统块”按钮，并双击之，如图 8-23 所示。




图 8-23　打开系统块

6）设置从站的 MPI 通信参数。先将用于 MPI 通信的接口（本例为 port0）的地址设置成“3”，一定不能设定为“2”，再将波特率设定为“187.5 Kbps”，这个数值与 S7－300 的波特率必须相等，最后单击“确认”按钮，如图 8-24 所示，这一步不少初学者容易忽略，其实这一步非常关键，因为各站的波特率必须相等，这是一个基本原则。系统块设置完成后，还要将其下载到 S7－200 中，否则通信是不能建立的。

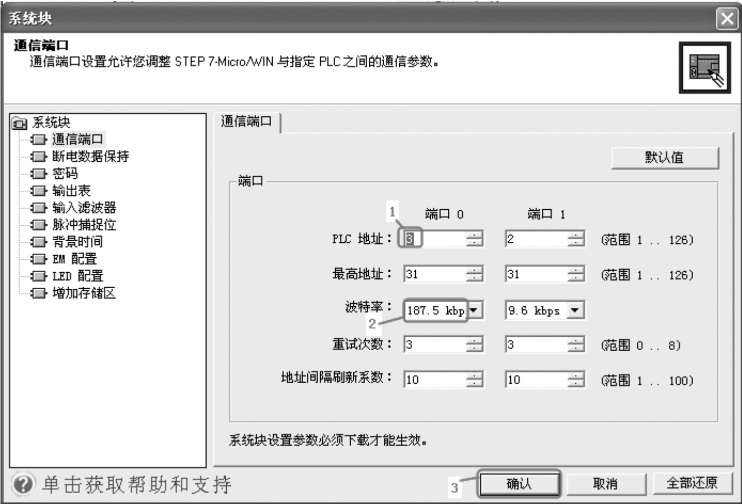



图 8-24　设置从站站的 MPI 通信参数

**【关键点】**硬件组态时，必须将 S7－200 PLC 和 S7－300 PLC 的波特率设置值应相等，此外 S7－300 PLC 的硬件组态和 S7－200 的系统块必须下载到相应的 PLC 中才能起作用。

**3. 相关指令介绍**

无组态连接的 MPI 的通信适合 S7－400、S7－300 以及 S7－200 之间的通信，通过调用 SFC66、SFC67、SFC68 和 SFC69 来实现。顾名思义，MPI 无组态连接就是 MPI 通信时，不

需要组态通信，只要编写通信程序即可实现通信。无组态连接的 MPI 通信分为双边编程通信方式和单边编程通信方式。S7 -200 与 S7 -300 间的 MPI 通信只能采用单边无组态通信方式。

X_PUT（SFC68）发送数据的指令，通过 SFC68 “X_PUT”，将数据写入不在同一个本地 S7 站中的通信伙伴。在通信伙伴上没有相应 SFC。在通过 REQ =1 调用 SFC 之后，激活写作业。此后，可以继续调用 SFC，直到 BUSY =0 指示接收到应答为止。

必须要确保由 SD 参数（在发送 CPU 上）定义的发送区和由 VAR_ADDR 参数（在通信伙伴上）定义的接收区长度相同。SD 的数据类型还必须和 VAR_ADDR 的数据类型相匹配。其输入和输出的含义见表 8-3。

**表 8-3　X_PUT（SFC68）指令格式**

| LAD | 输入/输出 | 含　　义 | 数 据 类 型 |
|---|---|---|---|
| "X_PUT"<br>EN　ENO<br>REQ　RET_VAL<br>CONT　BUSY<br>DEST_ID<br>VAR_ADDR<br>SD | EN | 使能 | BOOL |
| | REQ | 发送请求 | BOOL |
| | CONT | 作业结束之后是否保持建立与通信伙伴的连接 | BOOL |
| | DEST_ID | 对方的 MPI 地址 | WORD |
| | VAR_ADDR | 对方的数据区 | ANY |
| | SD | 本机的数据区 | ANY |
| | RET_VAL | 返回数值（如错误值） | INT |
| | BUSY | 发送是否完成 | BOOL |

X_GET（SFC67）接收数据的指令，通过 SFC67"X_GET"，可以从本地 S7 站以外的通信伙伴中读取数据。在通信伙伴上没有相应 SFC。在通过 REQ =1 调用 SFC 之后，激活读作业。此后，可以继续调用 SFC，直到 BUSY =0 指示数据接收为止。然后，RET_VAL 便包含了以字节为单位的、已接收的数据块的长度。

必须要确保由 RD 参数定义的接收区（在接收 CPU 上）至少和由 VAR_ADDR 参数定义的要读取的区域（在通信伙伴上）一样大。RD 的数据类型还必须和 VAR_ADDR 的数据类型相匹配。其输入和输出的含义见表 8-4。

**表 8-4　X_GET（SFC67）指令格式**

| LAD | 输入/输出 | 含　　义 | 数 据 类 型 |
|---|---|---|---|
| "X_GET"<br>EN　ENO<br>REQ　RET_VAL<br>CONT　BUSY<br>DEST_ID　RD<br>VAR_ADDR | EN | 使能 | BOOL |
| | REQ | 接收请求 | BOOL |
| | CONT | 作业结束之后是否保持建立与通信伙伴的连接 | BOOL |
| | DEST_ID | 对方的 MPI 地址 | WORD |
| | VAR_ADDR | 对方的数据区 | ANY |
| | RD | 本机的数据区 | ANY |
| | RET_VAL | 返回数值（如错误值） | INT |
| | BUSY | 接收是否完成 | BOOL |

**4. 程序编写**

X_PUT（SFC68）发送数据的指令和 X_GET（SFC67）接收数据的指令是系统功能，也就是系统预先定义的功能，只要将“库”展开，再展开“Standart libarary（标准库）”，选定“X_PUT”或者“X_GET”，再双击之，“X_PUT”或者“X_GET”就自动弹出到网络中指定的位置，如图 8-25 所示。

主站的梯形图如图 8-26 ~ 图 8-28 所示，从站的梯形图如图 8-29 所示，有时从站可以不编写梯形图程序。




图 8-25 X_PUT 和 X_GET 指令的位置

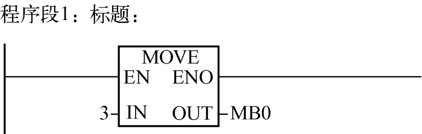
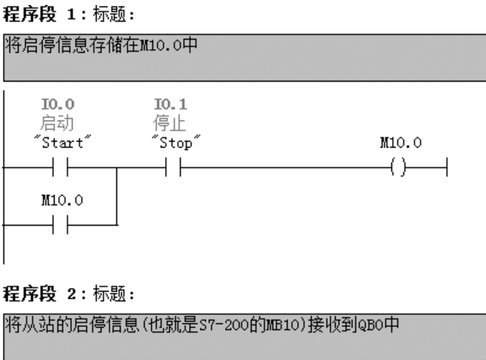



图 8-26 主站 OB100 中的梯形图




程序段 2：标题：

将从站的启停信息(也就是S7-200的MB10)接收到QB0中

SFC67
Read Data from a Communication Partner outside the Local S7 Station
"X_GET"
EN ENO
M0.0 REQ RET_VAL MW20
M0.1 CONT BUSY M22.0
W#16#3 DEST_ID RD P#Q 0.0 BYTE 1
P#M 10.0 BYTE 1 VAR_ADDR

图 8-27 主站 OB1 中的梯形图

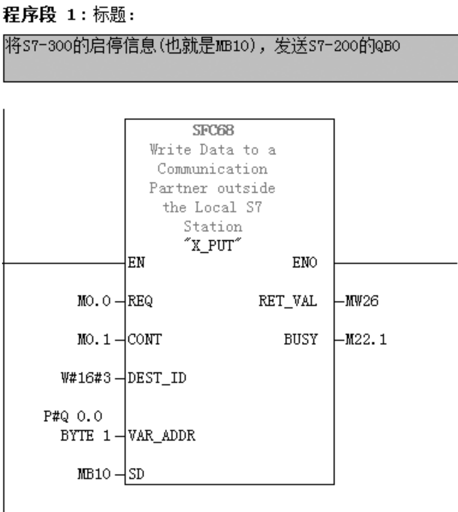



图 8-28 主站 OB35 中的梯形图

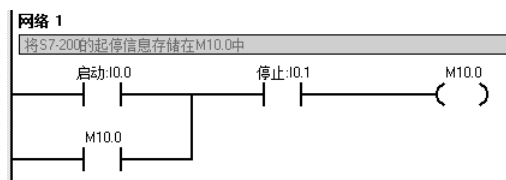



图 8-29　从站梯形图

【关键点】本例主站地址为“2”，从站的地址为“3”，因此硬件配置采用方案 1 时，必须将“PPI 口”的地址设定为“3”。而采用方案 2 时，必须将 EM277 的地址设定为“3”，设定完成后，还要将 EM277 断电，新设定的地址才能起作用。指令“X_PUT”的参数 SD 和 VAR_ADDR 的数据类型可以据实际情况确定，但在同一程序中数据类型必须一致。

【例 8-2】某设备上有一台 CPU226CN，CPU226CN 上的两个 PPI 接口已经被占用，现要改造此设备，需要添加一台 TP177B 触摸屏，请设计一个方案。

**解**：CPU226CN 上的两个 PPI 接口已经被占用，所以要建立 TP177B 触摸屏和 CPU226CN 的通信，就必须扩展 CPU226CN 通信接口。扩展一个 EM277 模块即可，TP177B 触摸屏和 CPU226CN 可以采用 MPI 或者 PROFIBUS 通信，硬件配置方案如图 8-30 所示。




图 8-30　硬件配置方案

**5. SFC 通信简介**

S7300/400 与 S7-200 的 MPI 通信是一种 SFC 通信，SFC 通信的特点如下：

① 使用 MPI 子网或者在一个站内进行数据交换。

② 与 SFB 通信相比，无须组态连接。

③ 与对方的连接是动态建立和断开的。

④ 它是一种小数据量的通信，最多可传输 76B 数据，这比后续介绍 SFB 通信的数据量要少得多。

## 8.3.3　S7-300 与 S7-300 间的 MPI 通信

S7-300 与 S7-300 间的 MPI 通信除了可以采用前述的无组态通信方式外，还可以采用全局数据通信方式，这种通信方式可以在 S7-300 与 S7-300、S7-400 与 S7-300 以及 S7-400 与 S7-400 之间通信，用户不需要编写程序，在硬件组态时组态所有 MPI 的 PLC 站之间的发送区与接收区即可。以下用一个例子介绍 S7-300 与 S7-300 之间的全局数据 MPI 通信。

【例 8-3】有两台设备，由 1 台 CPU313C-2DP 和 CPU314C-2DP 控制，要求实时从设备 1 上的 CPU313C-2DP 的 MB10 ~ MB14 发出 5B 到设备 2 的 CPU314C-2DP 的 MB10 ~ MB14，从设备 2 上的 CPU314C-2DP 的 MB30 ~ MB34 发出 5B 到设备 1 的 CPU313C-2DP 的 MB30 ~ MB34。

**解**：将设备 1 上的 CPU313C－2DP 作为主站，主站地址为 2，将设备 2 上的 CPU314C－2DP 作为从站，从站地址为 3。

**1. 主要软硬件配置**

① 1 套 STEP 7 V5.5 SP4。

② 1 台 CPU313C－2DP 和 1 台 CPU314C－2DP。

③ 1 根编程电缆。

④ 1 根 PROFIBUS 网络电缆（含两个网络总线连接器）。

MPI 通信硬件配置图如图 8-31 所示。

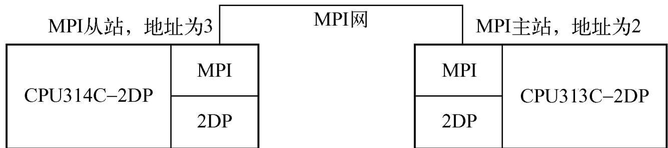



图 8-31　MPI 通信硬件配置图

**2. 硬件组态**

1）新建项目和插入站点。新建项目，本例的项目名为“MPI_300”，再在项目中插入两个站点，并重命名为“MASTER”和“SLAVE”，选定站点“MASTER”，双击“硬件”，如图 8-32 所示。




图 8-32　新建项目和插入站点

2）插入导轨。双击“Rail”，弹出导轨，如图 8-33 所示。注意此导轨编号为 0。




图 8-33　插入导轨

3）打开 CPU313C－2DP 属性。双击槽位 2 的“CPU 313C－2DP”，如图 8-34 所示。

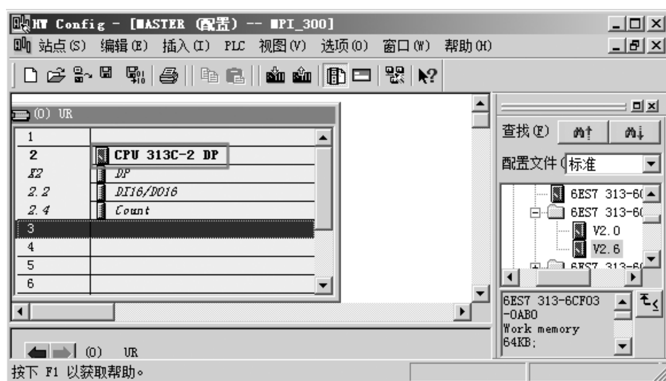



图 8-34　打开 CPU 314C－2DP 属性

4）设置站 2 的 MPI 通信参数。单击“属性”按钮，如图 8-35 所示，弹出设置 MPI 通信参数界面，如图 8-36 所示，设定 MPI 的地址为“2”，MPI 的通信波特率为“187.5 Kbps”，再单击“确定”按钮。



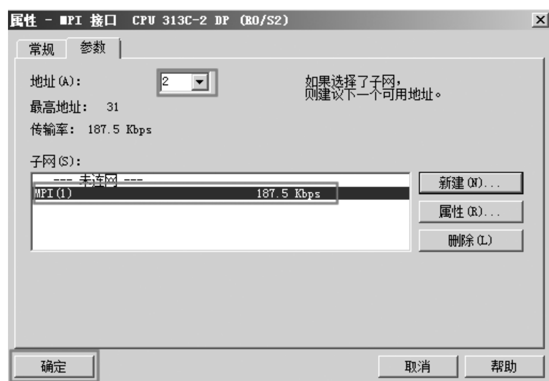




图 8-35　打开 MPI 通信参数设置界面　　　　图 8-36　设置站 2 的 MPI 通信参数

5）站 3 的硬件组态。回到图 8-32，选定“SLAVE”，双击“硬件”，弹出硬件组态界面，先插入导轨，再插入 CPU 模块，如图 8-37 所示。双击槽位 2 的“CPU 314C－2DP”。

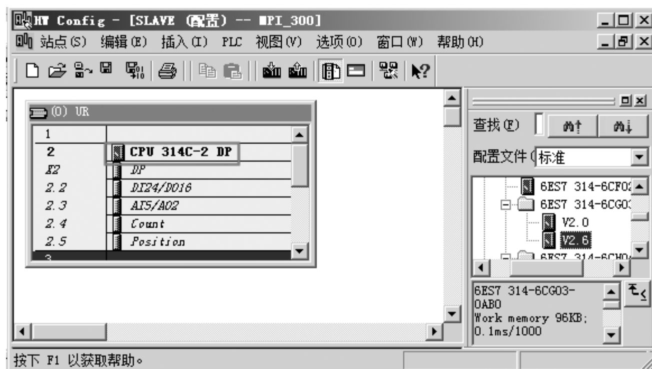



图 8-37　站 3 的硬件组态

6）打开 MPI 通信参数设置界面。单击“属性”按钮，如图 8-38 所示，弹出站 3 的 MPI 通信参数设置界面，如图 8-39 所示，设定 MPI 的地址为“3”，MPI 的通信波特率为“187.5 Kbps”，再单击“确定”按钮。

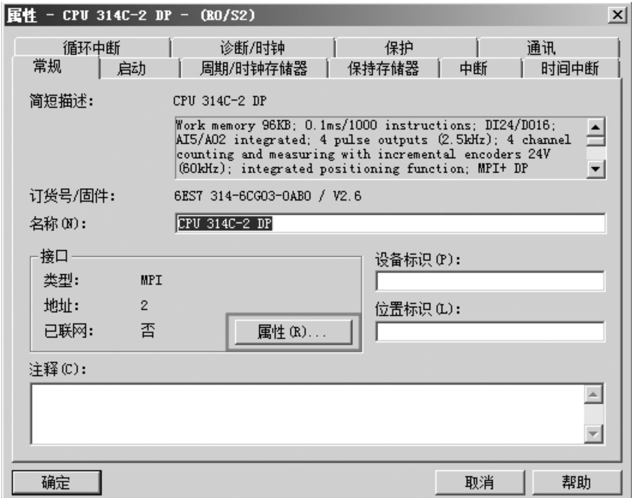



图 8-38　打开 MPI 通信参数设置界面

7）打开 MPI 网络。双击“MPI（1）”，如图 8-40 所示，弹出 MPI 的网络，如图 8-41 所示。




图 8-39　设置站 3 的 MPI 通信参数




图 8-40　打开 MPI 网络（1）

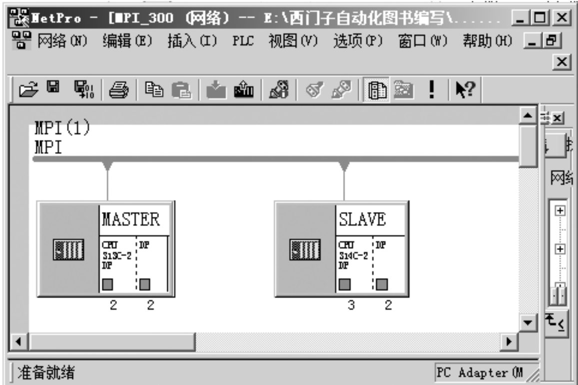



图 8-41　打开 MPI 网络（2）

8）打开全局变量发送、接收区组态。选中标记“1”处的“MPI（1）”网络线，再选中“选项”菜单，单击“定义全局数据”子菜单中的命令，打开全局变量发送、接收区组态，如图8-42所示。

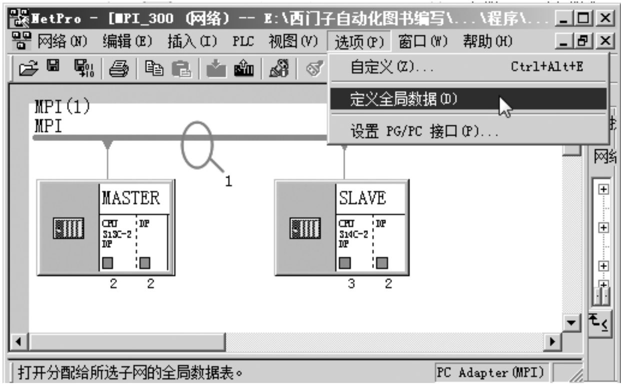



图8-42 打开全局变量发送、接收区组态

9）MPI全局变量组态。双击标记“1”处，如图8-43所示。

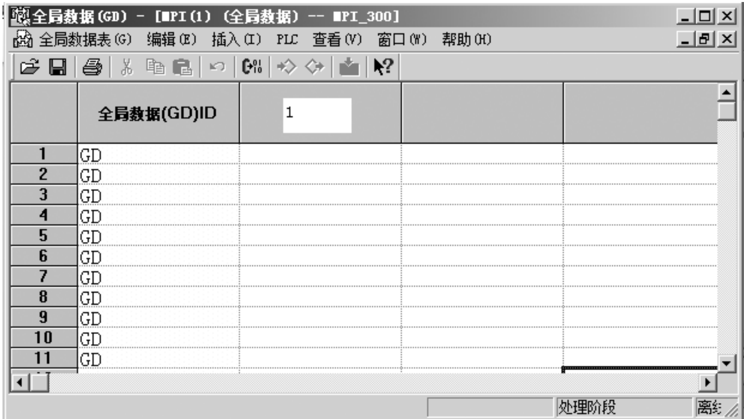



图8-43 MPI全局变量组态

10）选定CPU。选定“MASTER”，再选定“CPU 313C－2DP”，然后单击“确定”按钮，如图8-44所示。

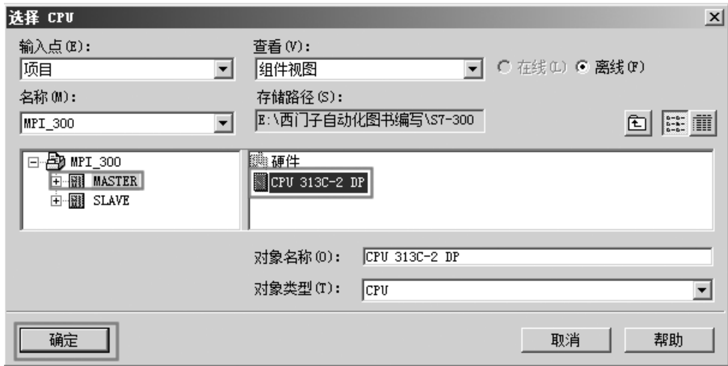



图8-44 选定CPU

11）定义发送区的数据组。输入 MB10:5，其含义是：将站点 MASTER MB10 开始的 5 个字节发送出去，如图 8-45 所示。

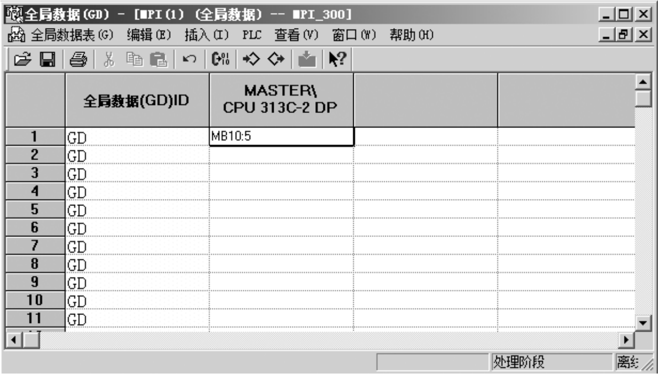



图 8-45　定义发送区的数据组

12）发送区的数据组的组态。选定“编辑”菜单，单击“发送器”子菜单，定义发送区的数据组，如图 8-46 所示，其他发送区和接收的数据组的组态方法类似，如图 8-47 所示。含义是：将站点 MASTER 从 MB10 开始的 5 个字节发送到 SLAVE 从 MB10 开始的 5 个字节的存储区中，将站点 SLAVE 从 MB30 开始的 5 个字节发送到 MASTER 从 MB30 开始的 5 个字节的存储区中。具体数据流向见表 8-5。




图 8-46　发送区的数据组的组态




图 8-47　发送区和接收的数据组的组态

表 8-5　全局 MPI 数据流向

| 序　　号 | MASTER | 对 应 关 系 | SLAVE |
|---|---|---|---|
| 1 | MB10 ~ MB14 | ⟶ | MB10 ~ MB14 |
| 2 | MB30 ~ MB34 | ⟵ | MB30 ~ MB34 |

13）保存和编译组态内容。单击“保存”按钮即可，如图 8-48 所示。

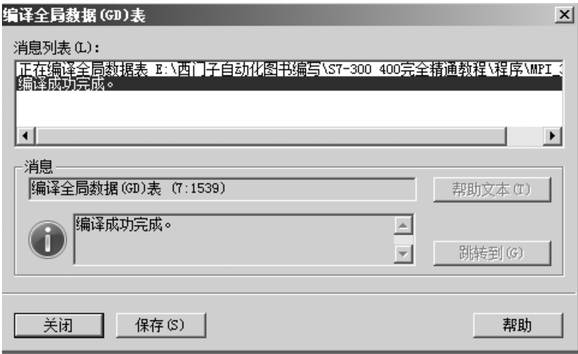



图 8-48　保存组态内容

14）下载组态信息。单击工具栏中的“下载”按钮，如图 8-49 所示。选定 MASTER 和 SLAVE 分别下载到对应的站点中去，如图 8-50 所示。

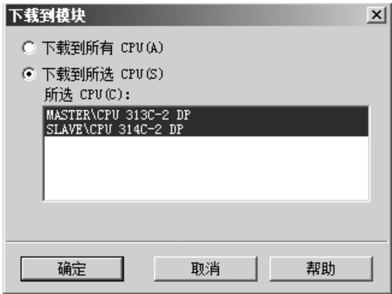



图 8-49　下载组态信息（1）




图 8-50　下载组态信息（2）

15）组态完成。经过编译后的界面如图 8-51 所示。GD X. Y. Z（如 GD 1. 2. 1）的含义见表 8-6。

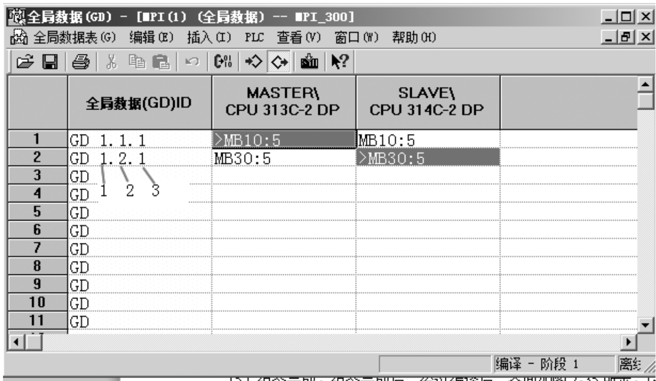



图 8-51　组态完成界面

表 8-6 GD X. Y. Z 的含义

| 序　　号 | 参　　数 | 含　　义 |
|---|---|---|
| 1 | X | 全局变量数据包的循环次数，次数与 CPU 有关，S7 - 300 最多支持 4 个 |
| 2 | Y | 一个循环中有几个数据包 |
| 3 | Z | 一个数据包中的数据区 |

**【关键点】** 本例的关键在于将 MPI 的通信组态正确，还有一点要特别注意，就是站 2 哪个数据区将数据送到站 3 哪个数据区中，站 2 又从站 3 哪个数据区接收数据，这些关系是绝对不能弄错的，否则不可能建立正确的通信。

## 8.3.4 S7 - 300/400 与 S7 - 400 间的 MPI 通信

在前面的章节讲解了无组态 MPI 通信、全局数据 MPI 通信，这些 MPI 的通信方式其实都适用于 S7 - 300/400 PLC 与 S7 - 400 间的 MPI 通信，但组态 MPI 通信方式就只适用于后者，以下用一个例子讲解 S7 - 300 与 S7 - 400 间的 MPI 通信。组态方式的 MPI 通信的好处是处理的数据量大。

**【例 8-4】** 有两台设备，由 1 台 CPU 416 - 2DP 和 CPU314C - 2DP 控制，要求实时从设备 1 上 CPU 416 - 2DP 的 MB10 ~ MB14 发出 5 个字节到设备 2 的 CPU314C - 2DP 的 MB10 ~ MB14，从设备 2 上的 CPU314C - 2DP 的 MB30 ~ MB34 发出 5 个字节到设备 1 的 CPU416 - 2DP 的 MB30 ~ MB34。

**解：** 将设备 1 上的 CPU 416 - 2DP 作为主站，主站地址为 2，将设备 2 上的 CPU 314C - 2DP 作为从站，从站地址为 3。

**1. 主要软硬件配置**

① 1 套 STEP 7 V5. 5 SP4。

② 1 台 CPU 314C - 2DP。

③ 1 台 CPU 416 - 2DP。

④ 1 根编程电缆。

⑤ 1 根 PROFIBUS 网络电缆（含两个网络总线连接器）。

MPI 通信硬件配置图如图 8-52 所示。

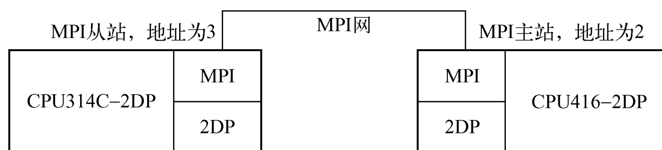



图 8-52 MPI 通信硬件配置图

**2. 硬件组态**

1）新建项目。新建项目，命名为“MPI_400”，插入站点和 CPU，并将建立 SLAVE 和 MASTER 的 MPI 连接，其中 MASTER 的 MPI 地址为“2”，SLAVE 的 MPI 地址为“3”，如图 8-53 所示，再单击“MPI（1）”标志，弹出如图 8-54 所示的界面。

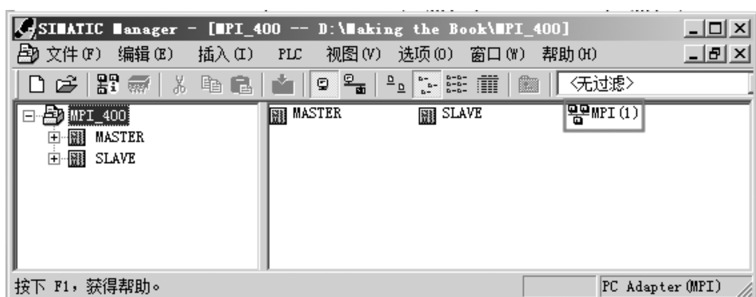



图 8-53　新建项目

2）新建连接。如图 8-54 所示，选中“1”处，单击鼠标右键，弹出快捷菜单，单击“插入新连接”命令，弹出如图 8-55 所示的界面。

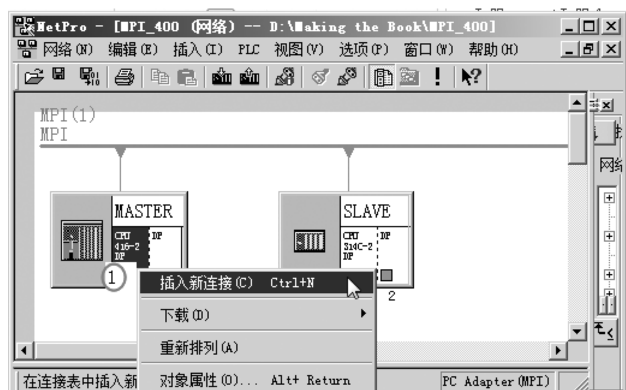



图 8-54　新建连接

3）选择 CPU 的连接方式。如图 8-55 所示，选中“CPU 314C－2DP”和“S7 连接”，单击“应用”按钮，弹出如图 8-56 所示的界面。

4）选择 MPI 块参数。如图 8-56 所示，单击“确定”按钮，硬件组态完成。

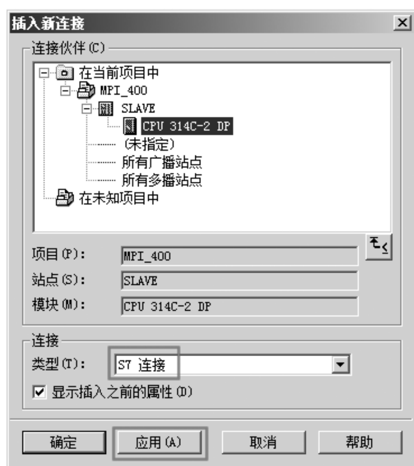



图 8-55　选择 CPU 的连接方式




图 8-56　块参数

**3. 相关指令介绍**

PUT（SFB15）是发送指令，通过使用 SFB15“PUT”，可以将数据写入到远程 CPU。在 REQ 的上升沿处发送数据。在 REQ 的每个上升沿处传送参数 ID、ADDR_1 和 SD_1。在每个作业结束之后，可以给 ID、ADDR_1 和 SD_1 参数分配新数值。其各参数的含义见表 8-7。

**表 8-7　PUT（SFB15）指令格式**

| LAD | 输入/输出 | 含　　义 | 数 据 类 型 |
|---|---|---|---|
| "PUT"<br>EN　ENO<br>REQ　DONE<br>ID　ERROR<br>ADDR_1　STATUS<br>ADDR_2<br>ADDR_3<br>ADDR_4<br>SD_1<br>SD_2<br>SD_3<br>SD_4 | EN | 使能 | BOOL |
| | REQ | 发送请求 | BOOL |
| | ID | 地址参数 | WORD |
| | ADDR_1 | 本地的存储地址 | ANY |
| | SD_1 | 对方的数据区 | ANY |
| | DONE | 是否发送完成 | BOOL |
| | ERROR | 是否错误 | BOOL |
| | STATUS | 状态 | WORD |

GET（SFB14）是接收指令，通过 SFB14“GET”，从远程 CPU 中读取数据。在 REQ 的上升沿处读取数据。在 REQ 的每个上升沿处传送参数 ID、ADDR_1 和 RD_1。在每个作业结束之后，可以分配新数值给 ID、ADDR_1 和 RD_1 参数。其各参数的含义见表 8-8。

**表 8-8　GET（SFB14）指令格式**

| LAD | 输入/输出 | 含　　义 | 数 据 类 型 |
|---|---|---|---|
| "GET"<br>EN　ENO<br>REQ　NDR<br>ID　ERROR<br>ADDR_1　STATUS<br>ADDR_2<br>ADDR_3<br>ADDR_4<br>RD_1<br>RD_2<br>RD_3<br>RD_4 | EN | 使能 | BOOL |
| | REQ | 接收请求 | BOOL |
| | ID | 地址参数 | WORD |
| | ADDR_1 | 本地的存储地址 | ANY |
| | RD_1 | 对方的数据区 | ANY |
| | NDR | 是否在接收完成 | BOOL |
| | ERROR | 是否错误 | BOOL |
| | STATUS | 状态 | WORD |

**【关键点】** PUT（SFB15）和 GET（SFB14）指令的参数 ID 设定如图 8-56 所示，本通信使用 OSI 模型的第 1、2 和 7 层。

**4. 编写程序**

主站的梯形图程序如图 8-57、图 8-58 所示。

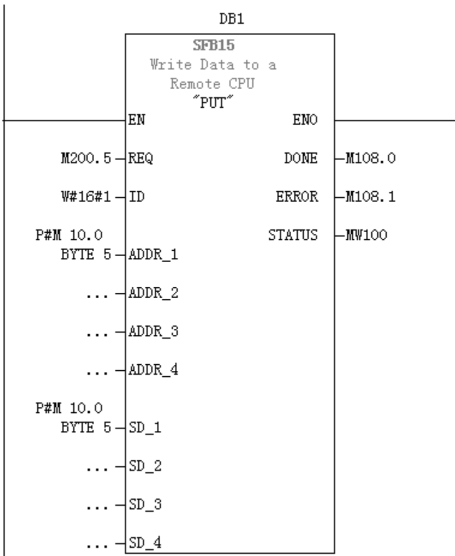



图 8-57　主站 OB35 梯形图

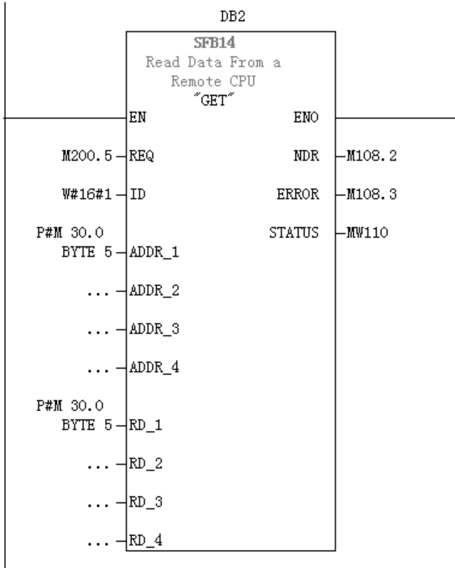



图 8-58　主站 OB1 梯形图

【关键点】从站中不需要编写程序，MPI、PROFIBUS－DP 以及以太网的 S7 通信与以上程序相同。

**5. SFB 通信简介**

SFB 块存在于所有的 S7－400 CPU 中，被用于同 S7/M7－300/400 CPU 交换数据。使用这些块，最多可有 64 KB 的数据通过多种子网（MPI、PROFIBUS、工业以太网）传输。SFB 通信有如下特点：

1）使用 MPI、K－Bus、PROFIBUS 或工业以太网进行数据交换。

2）通过连接表组态连接。

3）通过完全重新起动建立连接并使连接永久存在（即使是在 STOP 模式）。

4）用户数据大小可达 64 KB。

5）通信服务也可用于控制（停止、起动）通信伙伴。

6）SFB 只存在于 S7－400 CPU，FB 用于 S7－300。

7）通过一个连接可以处理不同的任务。

## 8.4　PROFIBUS 通信及其应用

### 8.4.1　PROFIBUS 通信概述

PROFIBUS 是西门子的现场总线通信协议，也是 IEC61158 国际标准中的现场总线标准

之一。现场总线 PROFIBUS 满足了生产过程现场级数据可存取性的重要要求，一方面它覆盖了传感器/执行器领域的通信要求，另一方面又具有单元级领域所有网络级通信功能。特别在“分散 I/O”领域，由于有大量的、种类齐全以及可连接的现场总线可供选用，因此 PROFIBUS 已成为事实上的国际公认标准。

**1. PROFIBUS 的结构和类型**

从用户的角度看，PROFIBUS 提供三种通信协议类型：PROFIBUS – FMS、PROFIBUS – DP 和 PROFIBUS – PA。

1）PROFIBUS – FMS（Fieldbus Message Specification，现场总线报文规范），使用了第一层、第二层和第七层。第七层（应用层）包含 FMS 和 LLI（底层接口）主要用于系统级和车间级的不同供应商的自动化系统之间传输数据，处理单元级（PLC 和 PC）的多主站数据通信。目前 PROFIBUS – FMS 已经很少使用。

2）PROFIBUS – DP（Decentralized Periphery，分布式外部设备），使用第一层和第二层，这种精简的结构特别适合数据的高速传送，PROFIBUS – DP 用于自动化系统中单元级控制设备与分布式 I/O（例如 ET 200）的通信。主站之间的通信为令牌方式（多主站时，确保只有一个起作用），主站与从站之间为主从方式（MS），以及这两种方式的混合。三种方式中，PROFIBUS – DP 应用最为广泛，全球有超过 3000 万的 PROFIBUS – DP 节点。

3）PROFIBUS – PA（Process Automation，过程自动化）用于过程自动化的现场传感器和执行器的低速数据传输，使用扩展的 PROFIBUS – DP 协议。

此外，对于西门子系统，PROFIBUS 提供了两种更为优化的通信方式，即 PROFIBUS – S7 通信和 S5 兼容通信。

① PROFIBUS – S7（PG/OP 通信）使用了第一层、第二层和第七层。特别适合 S7 PLC 与 HMI 和编程器通信，也可以用于 S7 – 300 和 S7 – 400 以及 S7 – 400 和 S7 – 400 之间的通信。

② PROFIBUS – FDL（S5 兼容通信）使用了第一层和第二层。数据传送快，特别适合 S7 – 300、S7 – 400 和 S5 系列 PLC 之间的通信。

**2. PROFIBUS 总线和总线终端器**

（1）总线终端器

PROFIBUS 总线符合 EIA RS485 标准，PROFIBUS RS85 的传输以半双工、异步及无间隙同步为基础的。传输介质可以是光缆或者屏蔽双绞线，电气传输每个 RS85 网段最多 32 个站点，在总线的两端为终端电阻，其结构如图 8-59 所示。

（2）最大电缆长度和传输速率的关系

PROFIBUS DP 段的最大电缆长度和传输速率有关，传输的速度越大，则传输的距离越近，对应关系如图 8-60 所示。一般设置通信波特率不大于 500 Kbit/s，电气传输距离不大于 400 m（不加中继器）。

（3）PROFIBUS – DP 电缆

PROFIBUS – DP 电缆是专用的屏蔽双绞线，外层为紫色。PROFIBUS – DP 电缆钢结构和功能如图 8-61 所示。外层是紫色绝缘层，编制网防护层主要防止低频干扰，金属箔片层为防止高频干扰，最里面是两根信号线，红色为信号正接总线连接器的第 8 引脚，绿色为信号负接总线连接器的第 3 引脚。PROFIBUS – DP 电缆的屏蔽层“双端接地”。

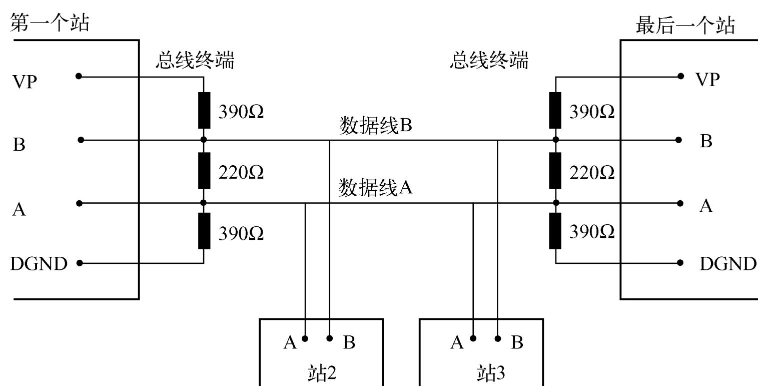


图 8-59　终端电阻的结构

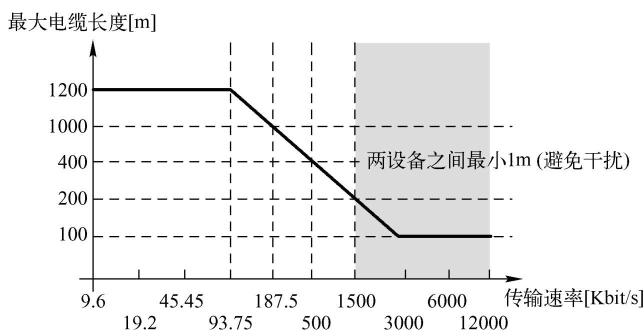


图 8-60　传输距离与波特率的对应关系

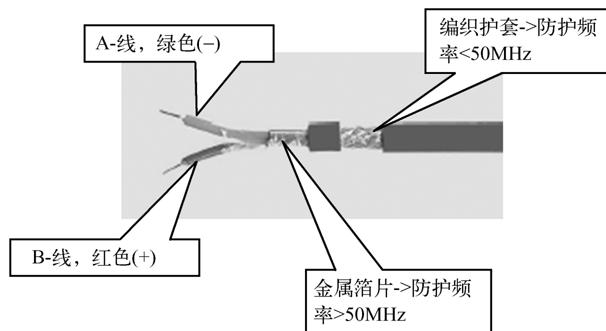


图 8-61　PROFIBUS－DP 电缆钢结构和功能

## 8.4.2　PROFIBUS 总线拓扑结构

### 1. PROFIBUS 电气接口网络

（1）RS－485 中继器的功能

如果通信的距离较远或者 PROFIBUS 的从站大于 32 个时，就要加入 RS－485 中继器。如图 8-52 所示，波特率为 500 Kbit/s 时，最大的传输距离为 400 m。如果传输距离大于 1000 m 时，需要加入两台 RS－485 中继器，就可以满足长度和传输速率的要求，拓扑结构如图 8-62 所示。

西门子的 RS－485 中继器具有信号放大和再生功能，在一条 PROFIBUS 总线上最多可以安装 9 台 RS－485 中继器。一个 PROFIBUS 网络的一个网段最多 32 个站点，如果一个 PRO-

FIBUS 网络多余 32 个站点就要分成多个网段，如一个 PROFIBUS 网络有 70 个站点，就需要两台 RS－485 中继器将网络分成 3 个网段。

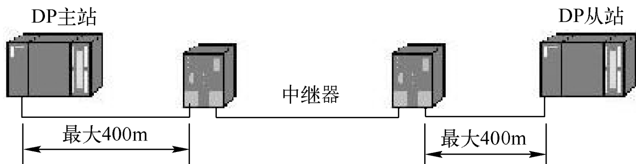



图 8-62　RS－485 中继器进行网络拓展

（2）利用 RS－485 中继器的网络拓扑

PROFIBUS 网络可以利用 RS－485 中继器组成“星形”总线结构和“树形”网络总线结构。“星形”总线结构如图 8-63 所示，“树形”网络总线结构如图 8-64 所示。

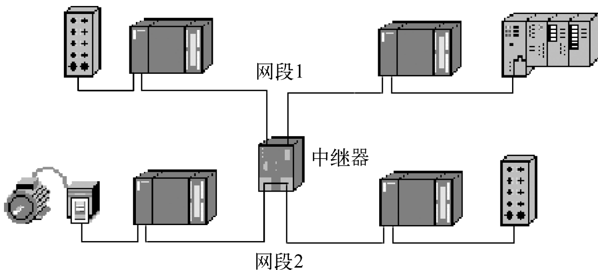



图 8-63　RS－485 中继器星形拓扑结构

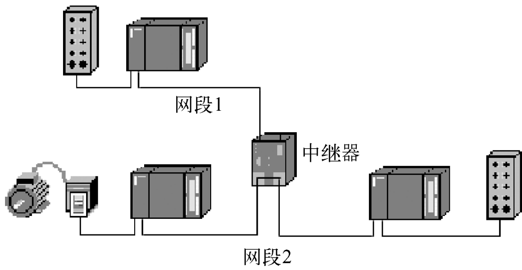



图 8-64　RS－485 中继器树形拓扑结构

**2. PROFIBUS 光纤接口网络**

对于长距离数据传输，电气网络往往不能满足要求，而光纤网络可以满足长距离数据传输且保持高的传输速率。此外，光纤网络有较好的抗电磁干扰能力。

利用光纤作为传输介质，把 PLC 接入光纤网络，有三种接入方式：

（1）集成在模块上的光纤接口

例如 CP342－5 FO、IM153－2 FO 和 IM467 FO，这些模块末尾都有“FO”标记。这些模块的光纤分为塑料光纤和 PCF 光纤。使用塑料光纤时，两个站点的最大传输距离为 50 m。使用 PCF 光纤时，西门子的光纤的长度有 7 个规格，分别是 50 m、75 m、100 m、150 m、200 m、250 m 和 300 m。两个站点的最大传输距离为 300 m。

（2）用 OBT 扩展 PROFIBUS 电气接口

只有电气接头可以通过 OBT（Optical Bus Terminal）连接一个电气接口到光纤网上。这是一种低成本的简易连接方式。但 OBT 只能用于塑料光纤和 PCF 光纤；一个 OBT 只能连接一个 PROFIBUS 站点；只能组成总线网，不能组成环网；因此应用并不多见。

（3）用 OLM 扩展 PROFIBUS 电气接口

如果普通的 PROFIBUS 站点设备没有光纤接头，只有电气接头可以通过 OLM（Optical Link Module）连接一个电气接口到光纤网上。OLM 光连模块的功能是进行光信号和电信号的相互转换，这种连接方式最为常见。OLM 光连模块根据连接介质分为如下几种：OLM/P11（连接塑料光纤）、OLM/P12（连接 PCF 光纤）、OLM/G11（连接玻璃光纤，一个电气接口和一个光接口）和 OLM/G12（连接玻璃光纤，一个电气接口和两个光接口）。OLM 光连模块外形如图 8-65 所示。



图 8-65　OLM 光连模块外形

（4）OLM 的拓扑结构

利用 OLM 进行网络拓扑可分为三种方式，总线结构、星形结构和冗余环网。总线拓扑结构如图 8-66 所示，OLM 上面的是电气接口，下面的是光纤接口。注意，在同一个网络中，OLM 模块的类型和光纤类型必须相同，不能混用。

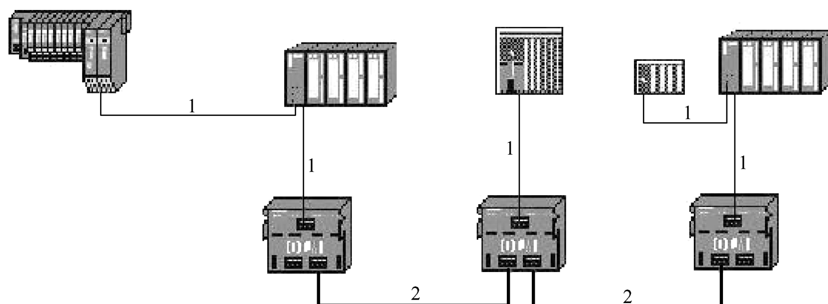


图 8-66　OLM 总线拓扑结构

1—RS-485 总线　2—光纤

总线拓扑结构简单，但如果一个 OLM 模块损坏或者光纤损坏，将造成整个网络不能正常工作，这是总线网络拓扑结构的缺点。

环形网络结构拓扑如图 8-67 所示，只是将如图 8-66 所示的首位相连，就变成冗余环形网络结构，其可靠性大为提高，在工程中应用较多。

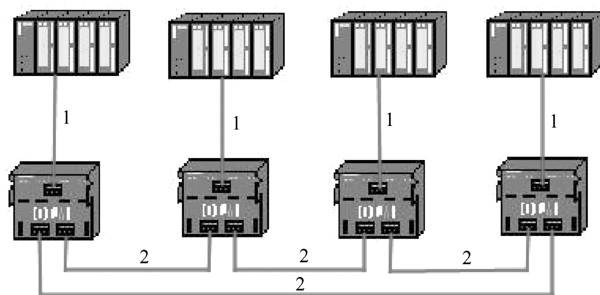


图 8-67　OLM 环形拓扑结构

1—RS-485 总线　2—光纤

星形网络结构拓扑如图 8-68 所示，其可靠性较高，但需要的投入相对较大。

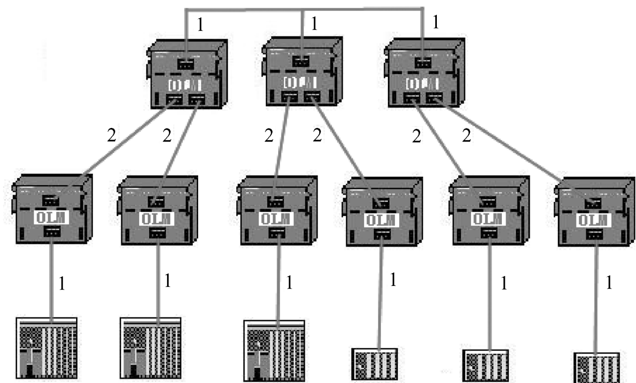



图 8-68　OLM 星形拓扑结构

1—RS485 总线　2—光纤

### 8.4.3　S7－300 与 ET200M 的 PROFIBUS－DP 通信

用 CPU314C－2DP 作为主站、分布式模块作为从站，通过 PROFIBUS 现场总线建立与这些模块（如 ET200S、EM200M 和 EM200B 等）的通信是非常方便的，这样的解决方案多用于分布式控制系统。这种 PROFIBUS 通信中，在工程中最容易实现，同时应用也最广泛。

**【例 8-5】** 有一台设备，控制系统由 CPU 314C－2DP、IM153－1 和 SM323 DI8×DO8 组成，请编写程序实现由主站 CPU 314C－2DP 发出一个起停信号控制从站一个中间继电器的通断；从站发出一个起停信号控制主站一个中间继电器的通断。

**解：** 将 314C－2DP 作为主站，将分布式模块作为从站。

**1. 主要软硬件配置。**

① 1 套 STEP 7 V5.5 SP4。

② 1 台 CPU 314C－2DP。

③ 1 台 IM153－1。

④ 1 块 SM323 DI8/DO8。

⑤ 1 根 PROFIBUS 网络电缆（含两个网络总线连接器）。

⑥ 1 根 PC/MPI 电缆。

PROFIBUS 现场总线硬件配置图如图 8－69 所示，PLC 和远程模块接线图如图 8－70 所示。

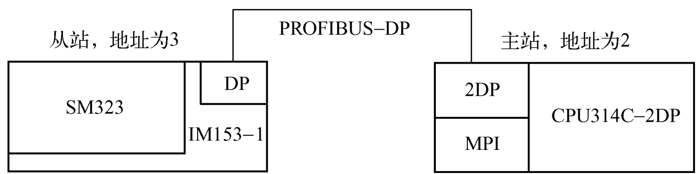



图 8-69　PROFIBUS 现场总线硬件配置图

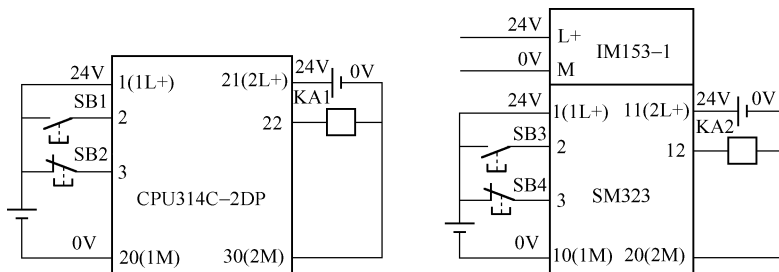


图 8-70　PROFIBUS 现场总线通信－PLC 和远程模块接线图

**2. 硬件组态**

1）新建项目和插入站点。先打开 STEP 7，再新建项目，本例命名为“ET200M”，接着单击菜单“插入”下的“站点”，并单击“SIMATIC 300 Station”，新建项目和插入主站如图 8-71 所示。

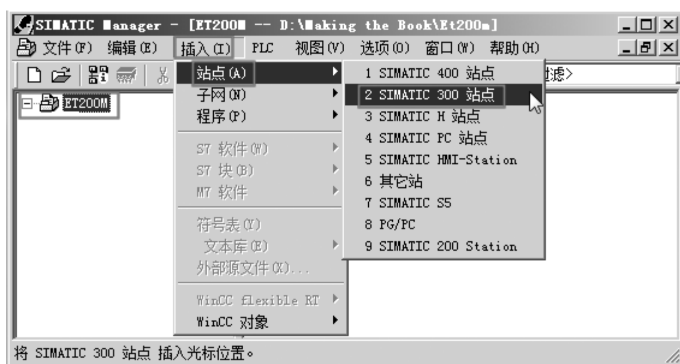


图 8-71　新建项目和插入站点

2）选中硬件。先单击“ET200M”前的“+”，展开“ET200M”，将“SIMATIC 300（1）”重命名为“Master”，再双击“硬件”，选中硬件如图 8-72 所示。



图 8-72　选中硬件

3）插入导轨。先单击 SIMATIC 300 前的“+”，展开 SIMATIC 300，再展开 Rack－300，然后双击“Rail”，弹出导轨 UR，如图 8-73 所示。

4）插入 CPU 模块。选中槽位 2，选中后槽位为绿色，展开 CPU－300，再展开 CPU 314－2DP，然后双击“V2.6”，如图 8-74 所示。

5）新建 PROFIBUS 网络。如图 8-75 所示，“地址”中的选项是主站的地址，本例确定为 2，再展开 CPU 314－2DP，再单击“新建”按钮，弹出如图 8-76 所示界面。单击“确

定”按钮，主站的 PROFIBUS 网络设定完成。




图 8-73　插入导轨




图 8-74　插入 CPU 模块







图 8-75　新建 PROFIBUS 网络（1）　　　　图 8-76　新建 PROFIBUS 网络（2）

6）修改主站的 I/O 地址。双击槽位“2.2”，弹出“属性”界面，取消“系统默认”前的“√”，再选定“地址”选项卡，并在“输入”的起始地址中输入 0，“输出”的起始地址也输入 0，最后单击“确定”按钮，修改主站的 I/O 地址如图 8-77 所示。



图 8-77　修改主站的 I/O 地址

7）将从站挂到 PROFIBUS 网络上。选中“1”处的 PROFIBUS 网络，再展开“PROFIBUS DP”下的“ET 200M”，并双击“IM 153－1”，组态从站硬件如图 8-78 所示。

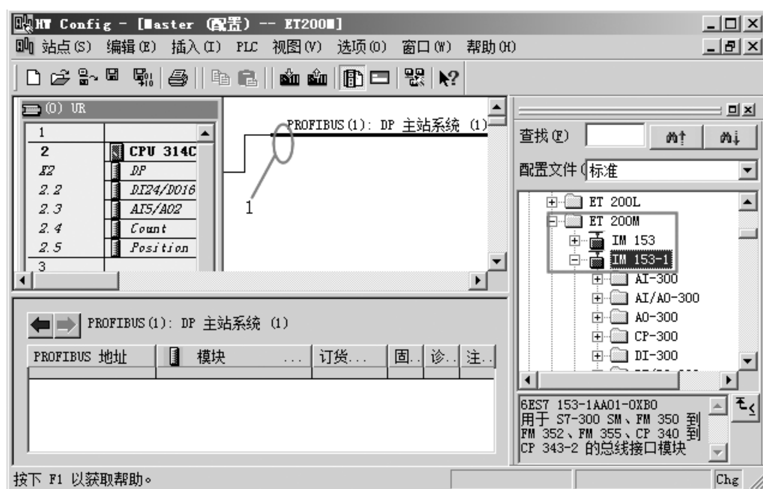


图 8-78　将从站挂到 PROFIBUS 网络上

8）设置从站地址。“地址”中的选项是从站的地址，本例设置为 3，再单击“确定”按钮，设置从站地址如图 8-79 所示。



图 8-79　设置从站地址

9）插入输入模块。选中“1”处的 IM153 - 1，再展开“DI/DO - 300”，并双击“SM 323 DI8/DO8x24V/0.5A ”，插入模块如图 8-80 所示。

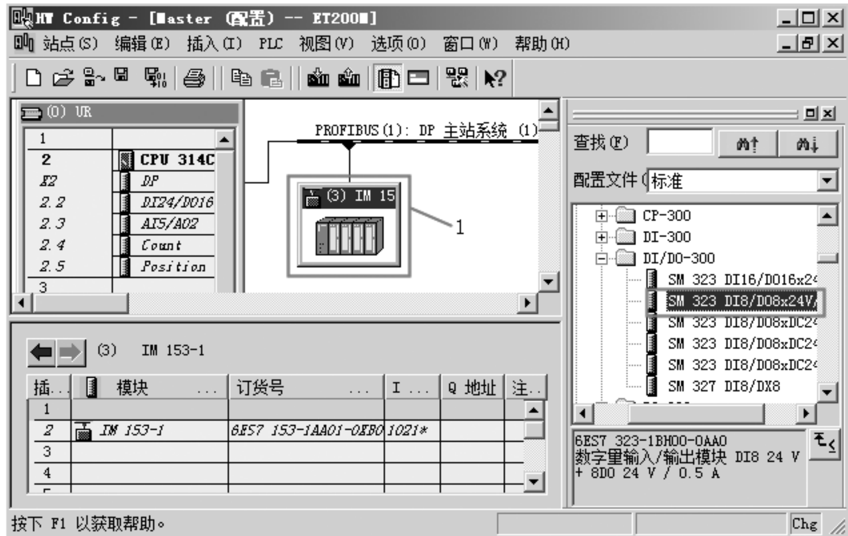



图 8-80　插入模块

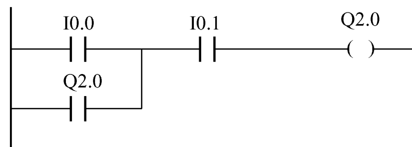
10）保存和编译硬件组态。单击工具栏的“保存和编译”按钮，对硬件组态进行编译，保存和编译如图 8-81 所示。从图中还可以看到：SM 323 的输入地址为 IB3，输出地址为 QB2。




图 8-81　保存和编译硬件组态

**3. 编写程序**

只需要对主站编写程序，主站的梯形图程序如图 8-82 所示。



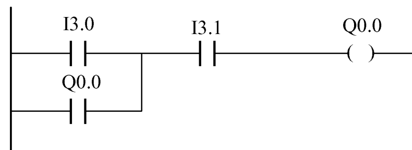




图 8-82　梯形图

### 8.4.4　S7-300 与 S7-200 间的 PROFIBUS-DP 通信

为了节约成本，在我国 S7-300 与 S7-200 间的现场总线通信在工业控制中的应用较常见，图 8-83 所示为某铜矿的 S7-300 与 S7-200 间 PROFIBUS-DP 现场总线通信硬件组态的实例，由于此实例比较复杂。在此不详细介绍。

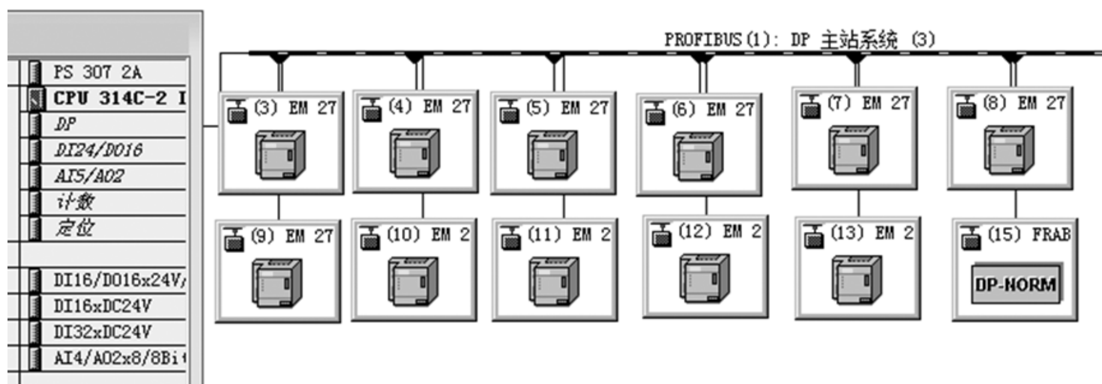



图 8-83　某铜矿 S7-300 与 S7-200 间 PROFIBUS-DP 现场总线通信硬件组态的实例

以下以一台 CPU 314C-2DP 与一台 CPU 226CN 之间 PROFIBUS 的现场总线通信为例，介绍 S7-200 与 S7-300 间的现场总线通信。

**【例 8-6】** 模块化生产线的主站为 CPU 314C-2DP，从站为 CPU 226CN 和 EM277 的组合，主站发出开始信号（开始信号为高电平），从站接收信息，并使从站的指示灯以 1 s 为周期闪烁。同理，从站发出开始信号（开始信号为高电平），主站接收信息，并使主站的指示灯以 1 s 为周期闪烁。

**解：**

**1. 主要软硬件配置**

① 1 套 STEP 7-Micro/WIN V4.0 SP9。

② 1 套 STEP 7 V5.5 SP4。

③ 1 台 CPU 226CN。

④ 1 台 EM277。

⑤ 1 台 CPU 314C－2DP。

⑥ 1 根编程电缆。

⑦ 1 根 PROFIBUS 网络电缆（含两个网络总线连接器）。

PROFIBUS 现场总线硬件配置如图 8-84 所示，PROFIBUS 现场总线通信 PLC 接线如图 8-85 所示。

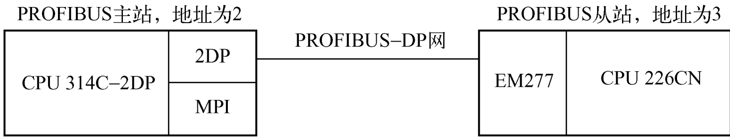



图 8-84　PROFIBUS 现场总线硬件配置图

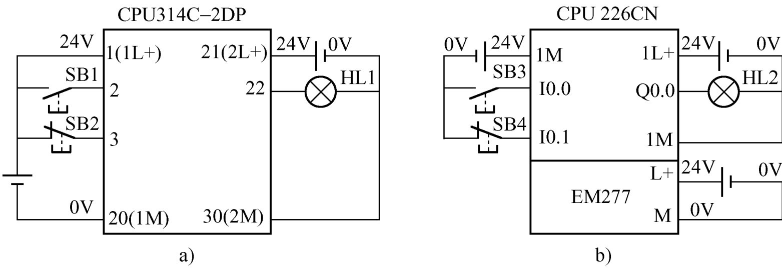



图 8-85　PROFIBUS 现场总线通信 PLC 接线图

a）CPU 314C－2DP PLC 接线图　b）CPU 226CN PLC 接线图

**2. CPU 314C－2DP 的硬件组态**

S7－300 与 S7－200 的 PROFIBUS 通信总的方法是：首先对主站 CPU 314C－2DP 的硬件进行组态，下载硬件，再编写主站程序，下载主站程序；编写从站程序，下载从站程序，最后便可建立主站和从站的通信。具体步骤如下：

1）打开 STEP 7 软件。双击桌面上的快捷键，打开 STEP 7 软件。当然也可以单击“开始”→“所有程序”→“SIMATIC”→“SIMATIC Manager”打开 STEP 7 软件。

2）新建项目。单击“新建”按钮，弹出“新建 项目”对话框，在“命名（M）”中输入一个名称，本例为“DP_200”再单击“确定”按钮，如图 8-86 所示。

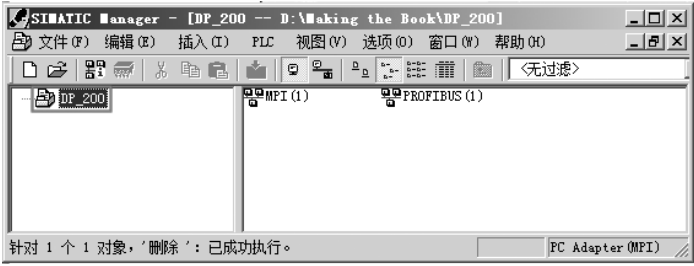



图 8-86　新建项目

3）插入站点。单击菜单栏“插入”菜单，再单击“站点”和“SIMATIC 300 站点”子菜单，如图 8-87 所示将主站“SIMATIC 300（1）”重命名为“Master”，双击“硬件”，打

开硬件组态界面，如图 8-88 所示。




图 8-87　插入站点

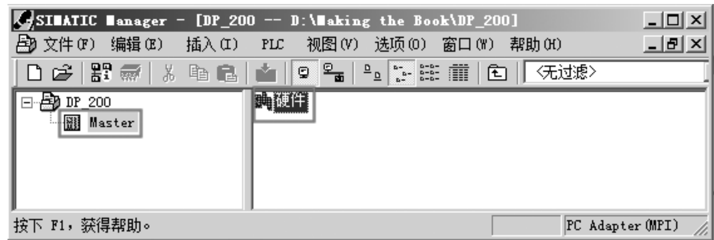



图 8-88　打开硬件组态

4）插入导轨。展开项目中的“SIMATIC 300”下的“RACK－300”，双击导轨“Rail”，如图 8-89 所示。硬件配置的第一步都是加入导轨，否则下面的步骤不能进行。




图 8-89　插入导轨

5）插入 CPU。展开项目中的“SIMATIC 300”下的“CPU－300”，再展开“CPU 314C－2DP”下的“6ES7 314－6CG06－OAB0”，将“V2.6”拖入导轨的 2 号槽中，如图 8-90 所示。若选用了西门子的电源，在配置硬件时应该将电源加入到第一槽，本例中使用的是开关电源，

因此硬件配置时不需要加入电源，但第一槽必须空缺，建议读者最好选用西门子电源。

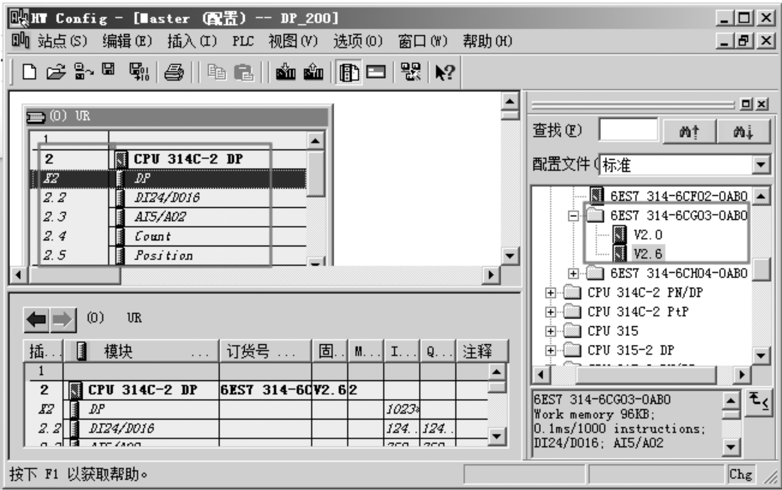



图 8-90　插入 CPU

6）配置网络。双击 2 号槽中的“DP”，弹出“属性 - DP”对话框，单击“属性”按钮，再弹出“属性 - PROFIBUS 接口”对话框，如图 8-91 所示；单击“新建”按钮，再弹出“属性 - 新建子网 PROFIBUS”对话框，如图 8-92 所示；选定传输率为“1.5 Mbps”和配置文件为“DP”，单击“确定”按钮，如图 8-93 所示。从站便可以挂在 PROFIBUS 总线上。




图 8-91　新建网络网络

7）修改 I/O 起始地址。双击 2 号槽中的“DI24/DO16”，弹出“属性 - DI24/DO16”对话框，如图 8-94 所示；去掉“系统默认”前的“√”，在“输入”和“输出”的“开始”中输入“0”，单击“确定”按钮，如图 8-95 所示。这一步骤目的主要是为了使程序中输入和输出的起始地址都从“0”开始，这样更加符合工程的习惯，若没有这个步骤，也是可行的，但程序中输入和输出的起始地址都从“124”开始。




图 8-92　设置通信参数

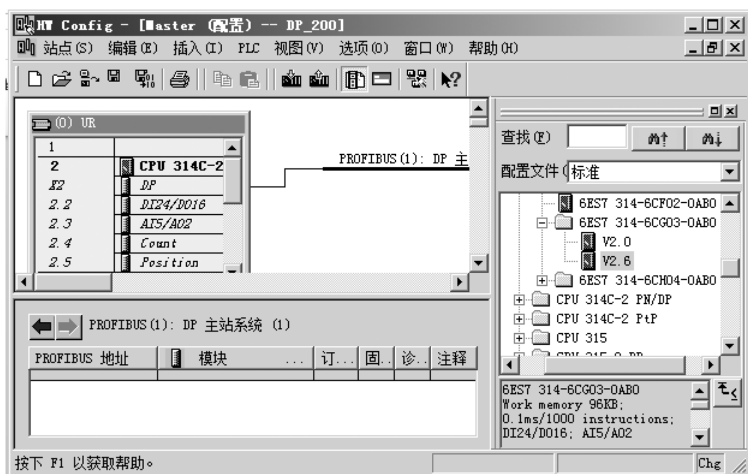



图 8-93　配置网络




图 8-94　修改 I/O 起始地址（1）




图 8-95　修改 I/O 起始地址（2）

8）配置从站地址。先选中“PROFIBUS”，再展开硬件目录，先后展开“PROFIBUS - DP”→“Additional Field Device”→“PLC”→“SIMATIC”，再双击“EM277 PROFIBUS - DP”，弹出“属性 - PROFIBUS 接口”对话框，将地址改为“3”，最后单击“确定”按钮，如图 8-96 所示。

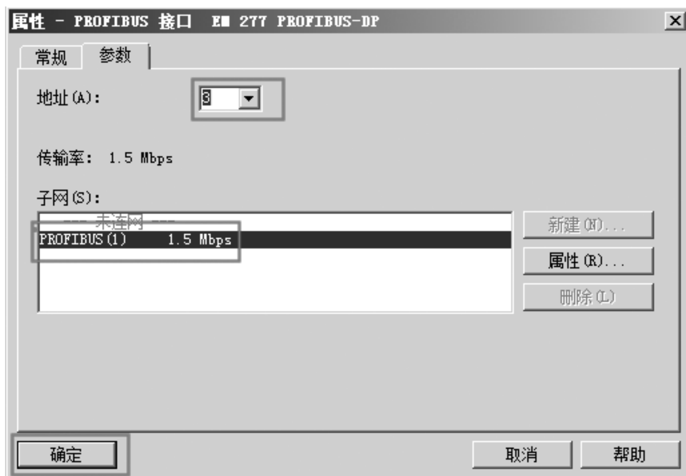



图 8-96　配置从站地址

9）分配从站通信数据存储区。先选中 3 号站，展开项目“EM277 PROFIBUS - DP”，再双击“2 Byte In/2 Byte Out”，如图 8-97 所示。当然也可以选其他的选项，这个选项的含义是：每次主站接收信息为 2 B，发送的信息也为 2 B。

10）设置周期存储器。双击“CPU 314C - 2DP”，打开属性界面，选中“周期/时钟存储器选项卡”，“勾选”时钟存储器，输入“100”，单击“确定”按钮即可，如图 8-98 所示。




图 8-97　分配从站通信数据存储区

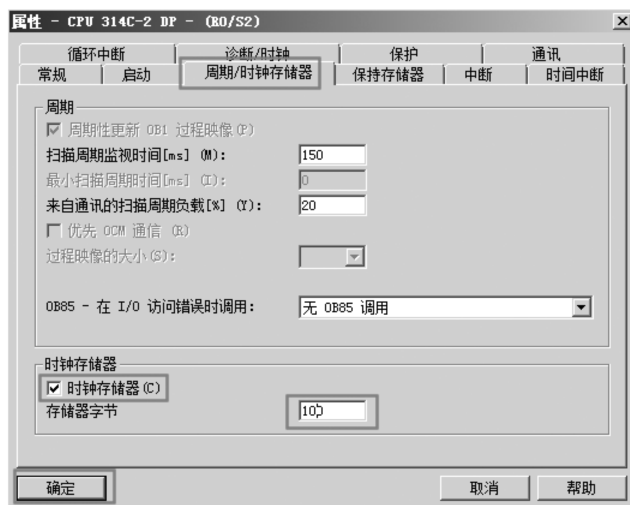



图 8-98　设置周期存储器

**3. 编写程序**

（1）编写主站的程序

按照以上步骤进行硬件组态后，主站和从站的通信数据发送区和接收数据区就可以进行数据通信了，主站和从站的发送区和接收数据区对应关系见表 8-9。

**表 8-9　主站和从站的发送区和接收数据区对应关系**

| 序　　号 | 主站 S7－300 | 对 应 关 系 | S7－200 从站 |
|---|---|---|---|
| 1 | QW2 | ⟶ | VW0 |
| 2 | IW3 | ⟵ | VW2 |

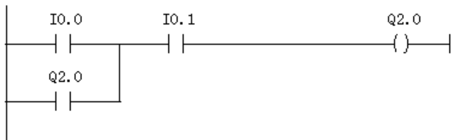
主站将信息存入 QW2 中，发送到从站的 VW0 数据存储区，那么主站的发送数据区为什么是 QW2 呢？因为 CPU 314C－2DP 自身是 16 点数字输出占用了 QW0，因此不可能是

QW0，QW2 是在前面的序 9）中设定的（就是步骤 9）。当然也可以设定为其他的单元，但不可以设定为 QW0。从站的接收区默认为 VW0，从站的发送区默认为 VW2，这个单元是可以在硬件组态时更改的，读者可参考西门子的相关手册。从站的信息可以通过 VW2 送到主站的 IW3。注意，务必要将组态后的硬件和编译后软件全部下载到 PLC 中。梯形图程序如图 8-99 所示。

（2）编写从站程序

在桌面上双击快捷键，打开软件 STEP 7 – Micro/WIN 软件，在梯形图中输入如图 8-100 的程序，再将程序下载到从站 PLC 中。

程序段 1：发送信息




程序段 2：接收信息

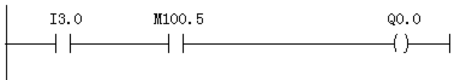



图 8-99　CPU 314C – 2DP 的程序

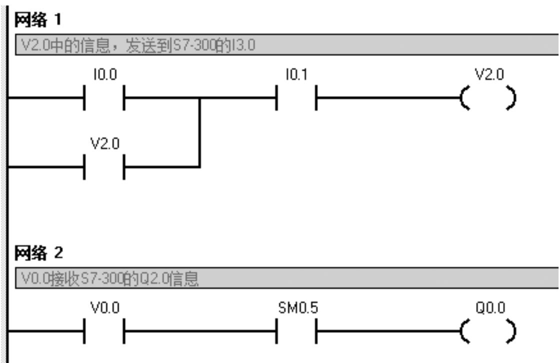



图 8-100　CPU 226CN 的程序

**4. 硬件连接**

主站 CPU 314C – 2DP 有两个 DB9 接口，一个是 MPI 接口，主要用于下载程序（也可作为 MPI 通信使用），另一个 DB9 接口是 DP 口，PROFIBUS 通信使用此接口。从站为 CPU 226CN + EM277，EM277 是 PROFIBUS 专用模块，这个模块的 DB9 接口为 DP 口。主站的 DP 口和从站的 DP 口由专用的 PROFIBUS 电缆和专用网络接头相连，主站和从站的硬件连线如图 8-84 所示。

PROFIBUS 电缆是二线屏蔽双绞线，两根线为 A 线和 B 线，电线塑料皮上印刷有 A、B 字母，A 线与网络接头上的 A 端子相连，B 线与网络接头上的 B 端子相连即可。B 线实际与 DB9 的第 3 针相连，A 线实际与 DB9 的第 8 针相连。

**【关键点】** 在前述的硬件组态中已经设定从站为第三站，因此在通信前，必须要将 EM277 的“站号”选择旋钮旋转到“3”的位置，否则，通信不能成功。此外，完成设定 EM277 的站地址后，必须将 EM277 断电，新设定的站地址才能生效。从站网络连接器的终端电阻应置于“on”，如图 8 – 101 所示。若要置于“off”，只要将拨钮拨向“off”一侧即可。

**5. 软硬件调试**

用 PROFIBUS 的电缆将 S7 – 300 的 DP 口与 EM277 的 DP 口相连，将 S7 – 300 端的网络连接器上的拨钮拨到“on”，将 EM277 端的网络连接器上的拨钮拨到“on”。再将程序下载

到 PLC 中。最后将两台 PLC 的运行状态从“STOP”都拨到“RUN”。

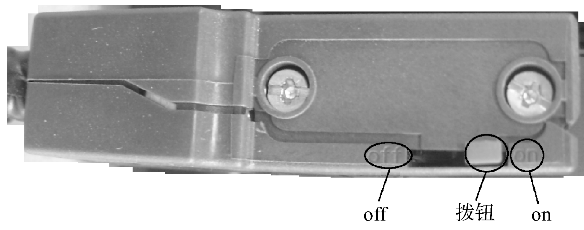



图 8-101　网络连接器的终端电阻置于“on”

### 8.4.5　S7－300 与 S7－300 间的 PROFIBUS－DP 通信

S7－300 与 S7－300 间的现场总线通信和 S7－300 与 S7－200 间的现场总线通信有所不同，有的 S7－300 CPU 自带 DP 通信口（如 CPU 314C－2DP），进行 PROFIBUS 通信时，只需要将两台 S7－300 CPU 的 DP 通信口用 PROFIBUS 通信电缆连接即可。而有的 S7－300 CPU 没有自带的 DP 通信口（如 CPU 314C），要进行 PROFIBUS 通信时，还必须配置 DP 接口模块（CP342－5）。以下以两台 CPU 314C－2DP 之间的 PROFIBUS 通信为例，介绍 S7－300 与 S7－300 间的 PROFIBUS 现场总线通信。

**【例 8-7】** 有两台设备，分别由一台 CPU 314C－2DP 控制，要求实时从设备 1 上的 CPU314C－2DP 的 MB10 发出 1 个字节到设备 2 的 CPU314C－2DP 的 MB10，从设备 2 上的 CPU314C－2DP 的 MB20 发出 1 个字节到设备 1 的 CPU314C－2DP 的 MB20。

**解：**

**1. 主要软硬件配置**

① 1 套 STEP 7 V5. 5 SP4。

② 两台 CPU 314C－2DP。

③ 1 根编程电缆。

④ 1 根 PROFIBUS 网络电缆（含两个网络总线连接器）。

PROFIBUS 现场总线硬件配置图如图 8-102 所示。

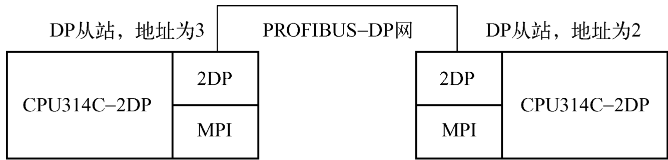



图 8-102　PROFIBUS 现场总线硬件配置图

**2. 硬件组态**

1）新建项目并插入站点。首先新建一个项目，本例为“Profibus－s7300”，如图 8-103 所示。再在项目中插入两个站点，本例为“Client”和“Server”，共插入两个站点，并将站点重命名为“Client”和“Server”，如图 8-104 所示。

2）插入导轨。如图 8-104 所示，选中从站“Client”，双击“硬件”，弹出如图 8-105 所示的界面，双击导轨“Rail”，弹出“1”处的导轨。




图 8-103　新建项目并插入站点

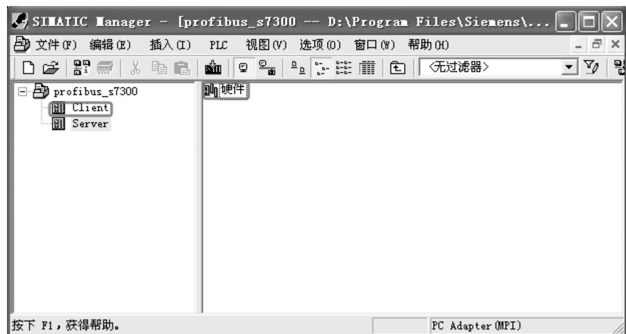



图 8-104　插入站点并重命名

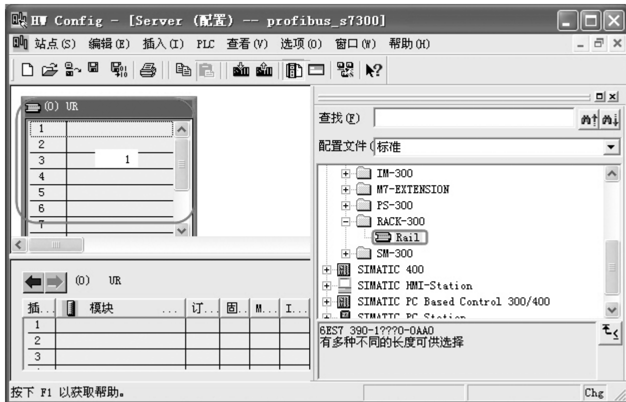



图 8-105　插入导轨

3）插入 CPU 模块。如图 8-106 所示，先选中导轨的 2 号槽位，再展开 CPU 314C-2DP，双击“V2.6”，也可直接用鼠标的左键选中“V2.6”并按住左键不放，直接将 CPU 拖曳到 2 号槽。

**【关键点】** CPU 314C-2DP 有多个产品型号，读者在组态时，要注意 CPU 314C-2DP 机壳上印刷的产品型号应与组态选择的产品型号一致。另外，“314-6CG03-0AB0”还有两个版本，在组态时也要注意与机壳上印刷的一致，否则会出错。

4）新建 PROFIBUS 网络。如图 8-107 所示，先设置从站的站地址为“3”，再单击“新建”按钮，弹出如图 8-108 所示的界面。




图 8-106 插入 CPU 模块




图 8-107 新建 PROFIBUS 网络

5）选择通信的波特率。在图 8-108 中，先选定 PROFIBUS 的通信的波特率为“1.5 Mbps”，再单击“确定”按钮，弹出如图 8-109 所示的界面。

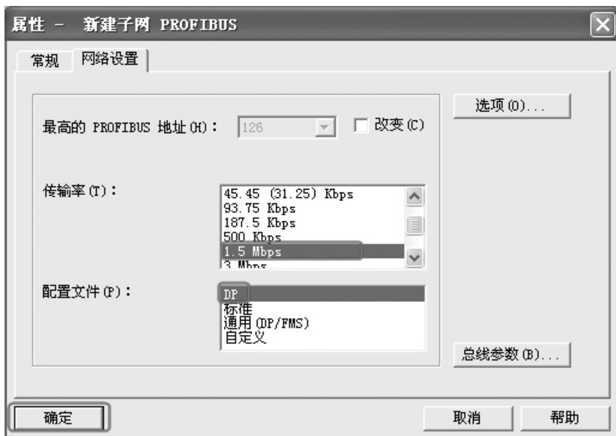



图 8-108 选择通信的波特率

6）选择操作模式。在图 8-109 中，先双击“1”处的 DP，再选择操作模式为“DP 从站”模式选项，再选定“组态”选项卡，弹出如图 8-110 所示的界面。

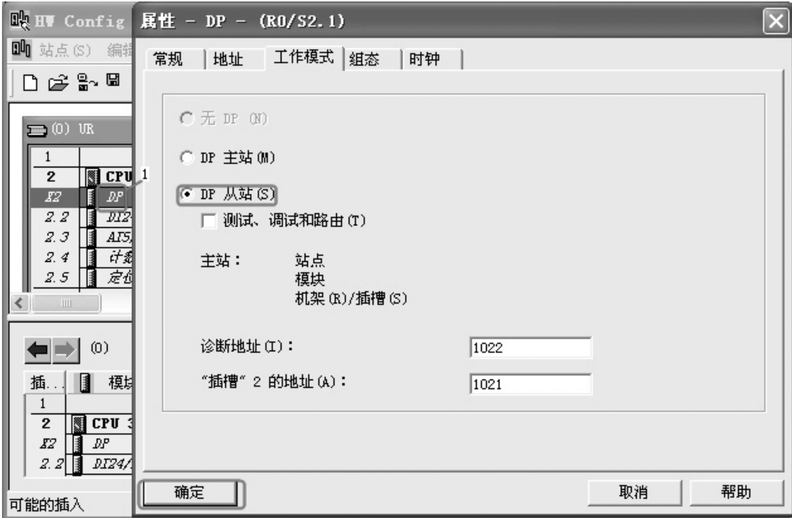



图 8-109　工作模式选择

7）组态接收区和接收区的数据。在图 8-110 中，单击“新建”按钮，弹出如图 8-111 所示的界面，定义从站 3 的接收区的地址为“3”（实际就是 IB3），单击“确定”按钮，接收区数据定义完成。再单击图 8-110 中的“新建”按钮，弹出如图 8-112 所示的界面，定义从站 3 的发送区的地址为“3”（实际就是 QB3），再单击“确定”按钮，发送区数据定义完成。弹出如图 8-113 所示的界面，单击“确定”按钮，从站的发送接收区数据组态完成。




图 8-110　组态通信接口数据区




图 8-111　组态接收区数据




图 8-112　组态发送区数据

8）主站组态时插入导轨和插入 CPU 与从站组态类似，这里不再重复，以下从选择通信波特率开始讲解。如图 8-114 所示，先选定主站 2 的通信地址为“2”，再选定通信的波特率为“1.5 Mbps”，单击“确定”按钮，弹出如图 8-115 所示的界面。

9）将从站 3 挂到 PROFIBUS 网络上。在图 8-115 中，先用鼠标选中 PROFIBUS 网络的“1”处，再双击“CPU 31x”，弹出如图 8-116 所示的界面。




图 8-113　从站数据区组态完成




图 8-114　选择通信波特率




图 8-115　将从站 3 挂到 PROFIBUS 网络上

10）激活从站 3。在图 8-116，单击“连接”按钮，弹出如图 8-117 所示的界面。




图 8-116　激活从站 3

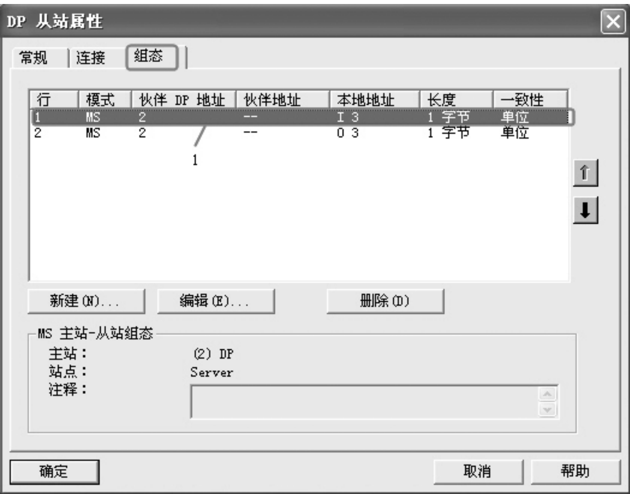



图 8-117　组态主站通信接口数据区

11）组态主站通信接口数据区。在 8-117 中，选中“组态”选项卡，双击“1”处，弹出如图 8-118 所示的界面。先选择地址类型为发送数据，再选定地址为“3”（实际就是 QB3），单击“确定”按钮，发送数据区组态完成。接收数据区的组态方法类似，只需要将如图 8-119 中地址类型选择为接收数据，再选定地址为“3”（实际就是 IB3），单击“确定”按钮。

12）硬件组态完成。在图 8-120 中，单击”确定”按钮，弹出如图 8-121 所示的界面。至此，主站的组态已经完成，

**【关键点】** 在进行硬件组态时，主站和从站的波特率要相等，主站和从站的地址不能相同，本例的主站地址为 2，从站的地址为 3。最为关键的是：先对从站组态，再对主站进行组态。

图 8-118　组态发送数据区

图 8-119　组态接收数据区

图 8-120　硬件组态完成（1）

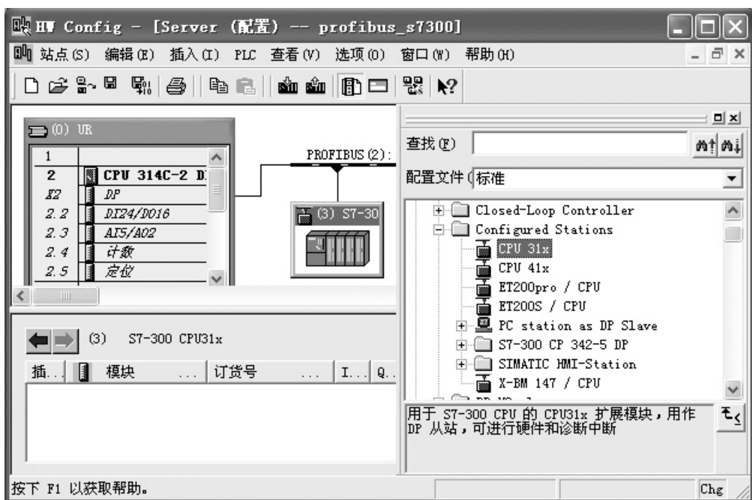



图 8-121 硬件组态完成（2）

**3. 编写主站程序**

S7-300 与 S7-300 间的现场总线通信的程序编写有很多种方法，本例是最为简单的一种方法。从图 8-120 中，很容易看出主站 2 和从站 3 的数据交换的对应关系，也可参见表 8-10。

**表 8-10 主站和从站的发送接收数据区对应关系**

| 序 号 | 主站 S7-300 | 对 应 关 系 | 从站 S7-300 |
|---|---|---|---|
| 1 | QB3 | ⟶ | IB3 |
| 2 | IB3 | ⟵ | QB3 |

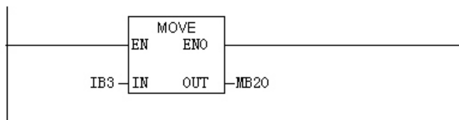
主站的程序如图 8-122 所示。

**4. 编写从站程序**

从站程序如图 8-123 所示。

**程序段 1：标题：**

主站接收从站QB3信息到IB3,之后传送到MB20




**程序段 2：标题：**

主站的MB10传送给QB3,QB3发送到从站的IB3

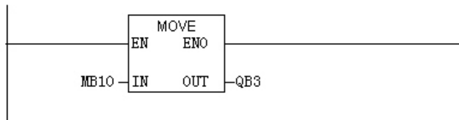
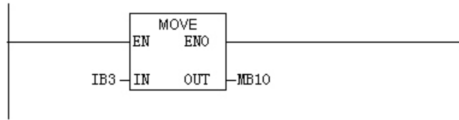



图 8-122 主站程序

**程序段 1：标题：**

从站把主站发送的信息接收到IB3，之后传送到MB10中




**程序段 2：标题：**

从站把MB20传送到QB3，再发送到主站的IB3中

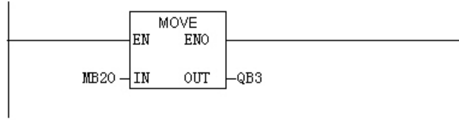



图 8-123 从站程序

### 8.4.6 CP342－5 作主站的 PROFIBUS－DP 通信

**1. PROFIBUS 通信处理器简介**

大多数 S7－300/400 CPU 都集成了 DP 接口。与使用 PROFIBUS 通信处理器相比，带集成 DP 接口的 CPU 的硬件成本要低很多。但 PROFIBUS 通信处理器扩展了 PLC 的通信接口，除此之外，PROFIBUS 通信处理器的功能比集成 DP 接口的功能更强。所有的 PROFIBUS 通信处理器都支持 PG/PC（编程器/操作面板）通信、S7 通信和 S5 兼容通信（FDL 通信），PROFIBUS 通信处理器还有故障诊断功能。

PROFIBUS 通信处理器主要有如下种类：

1）扩展 S7－300 通信接口的通信处理器，有 CP342－5、CP343－5 和 CP342－5 FO 三大类。CP342－5 用于 PROFIBUS 与 DP 协议的 S7 CP；CP342－5 FO 与 CP342－5 功能一样，只是有光纤接口；CP343－5 用于 PROFIBUS 与 FMS 协议的 S7 CP。

2）扩展 S7－400 通信接口的通信处理器，有 CP443－5Basic 和 CP4435 Extended 两类。CP443－5Basic 用于 PROFIBUS 与 FMS 协议的 S7 CP；CP4435 Extended 用于 PROFIBUS 与 DP 协议的 S7 CP。

CP342－5 可以作主站，也可以作从站，不管作主站还是作从站都只能安装在 S7－300 的中央机架上使用，不能在分布式从站（如 ET200M）上使用。

如果只是用 CP342－5 连接上位机或者操作面，这种情况使用 S7 协议。CP342－5 进行 PROFIBUS－DP 通信时，需要调用 FC1（DP_SEND）和 FC2（DP_RECV）。如果选择“NO DP”模式，不需要调用 FC1 和 FC2。

CP342－5 作主站时，最多可以连接 124 个从站，与每个从站最多可以交换 244 输入字节和 244 输出字节，与所有的从站总共最多交换 2160 个输入字节和 160 个输出字节。CP342－5 作从站时，与主站最多可以交换 240 输入字节和 240 输出字节。CP342－5 最多可以连接 16 个操作面板。

**2. CP342－5 作主站的 PROFIBUS－DP 通信应用**

以下介绍 CP342－5 作主站时，与 ET200M 之间的 PROFIBUS 通信的实例。

**【例 8-8】**有 1 台设备，控制系统由 CPU 314C－2DP、CP342－5、IM13－1 和 SM323 组成，要求实时从 CPU314C－2DP 的 MB0 发出 1 个字节到从站的 ET200M 上的 SM323，从从站的 ET200M 上的 SM323 接收 1 个字节到 CPU314C－2DP 的 MB10 中。

**解：**

（1）主要软硬件配置

① 1 套 STEP 7 V5.5 SP4。

② 1 台 CPU 314C－2DP。

③ 1 台 CP342－5。

④ 1 台 IM13－1 和 SM323。

⑤ 1 根编程电缆。

⑥ 1 根 PROFIBUS 网络电缆（含两个网络总线连接器）。

PROFIBUS 现场总线硬件配置图如图 8-124 所示。




图 8-124　PROFIBUS 现场总线硬件配置图

（2）硬件组态

1）新建项目并硬件组态。首先新建一个项目，本例为“CP_MASTER”，再插入 S7 - 300 站点；打开硬件组态界面，插入机架，再插入 CPU 314C - 2DP，如图 8-125 所示。

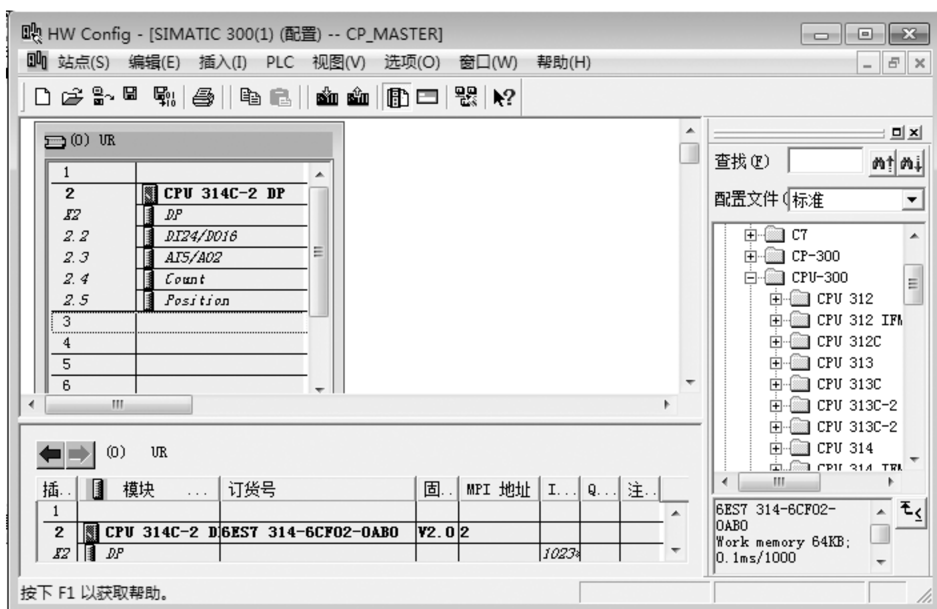



图 8-125　新建项目并插入站点

2）插入 CP342 - 5，并设置工作模式。在 4 号插槽中插入 CP342 - 5 通信处理器，双击 4 号插槽“CP342 - 5”，弹出“属性”对话框，如图 8-126 所示，选中“工作模式”选项卡，选择“DP 主站”工作模式，再单击“确定”按钮。弹出“属性 - PROFIBUS”对话框，单击“新建”按钮，实际就是新建 PROFIBUS 网络，如图 8-127 所示，设置主站地址为“2”，单击“确定”按钮。

3）组态从站。先选中如图 8-128 所示的 PROFIBUS 网线（标记为“1”处），选定硬件组态的目录树下“PROFIBUS DP”→“DP V0 Slaves”→“ET200M（IM153 - 1）”，按住鼠标左键，将其拖入如图 8-128 所示的网线位置，再松开鼠标。此时弹出“属性 - PROFIBUS”对话框，如图 8-129 所示，设置从站的地址为“3”，单击“确定”按钮。

选中拖入的分布式模块“ET200M”，插入混合模块 SM323，如图 8-130 所示，可以看到 SM323 的输入地址为 IB0，输出地址为 QB0。这两个字节用于和 CP342 - 5 模块交换数据（通信）。单击工具栏的“保存和编译”按钮，如没有显示错误，硬件组态工作完成。




图 8-126　设置站点为主站模式




图 8-127　新建 PROFIBUS 网络

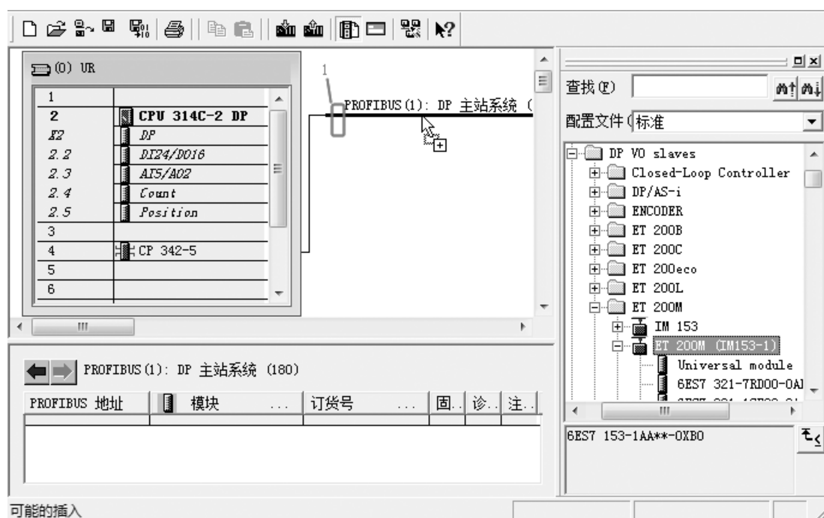



图 8-128　组态从站（1）




图 8-129　组态从站（2）

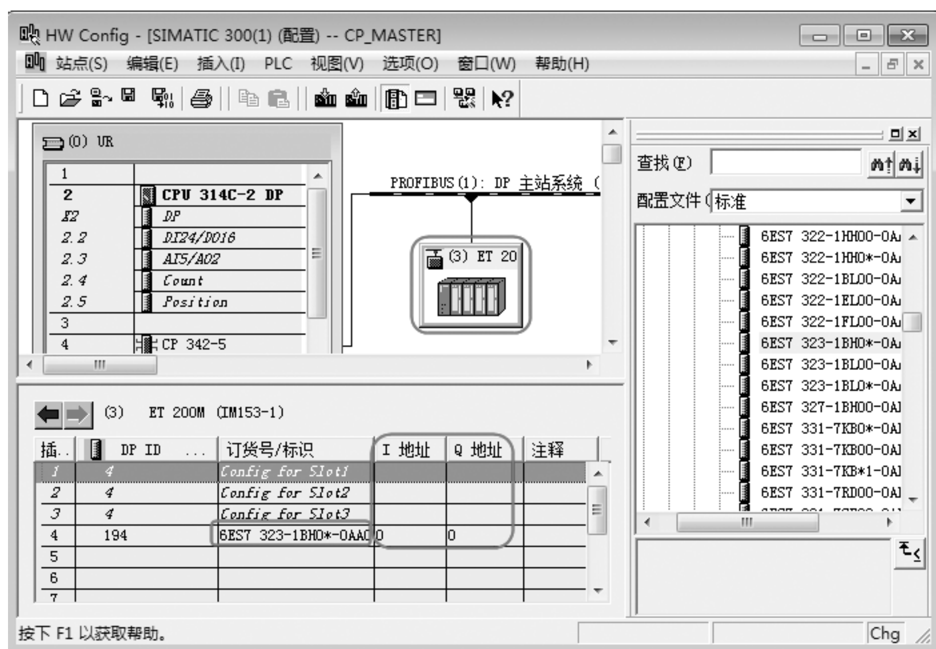



图 8-130　组态从站（3）

4）编写程序。与 CPU 集成的 DP 接口不同，CP342－5 作主站时，不能通过 I 和 Q 区直接交换数据到 ET200 的 I/O 中，需要调用 FC1（DP_SEND）和 FC2（DP_RECV），建立虚拟通信口区访问从站。

CP342－5 有一个内部的输入缓冲区和输出缓冲区，用于存放所有 DP 从站的 I/O 数据。PROFIBUS 从站自动将数据与 CP342－5 的缓冲区数据进行交换，并不需要编写程序，但 PLC 与 CP342－5 的缓冲区的数据交换需要调用 FC1（DP_SEND）和 FC2（DP_RECV）。本

例的主站与从站之间的数据交换如图 8-131 所示。

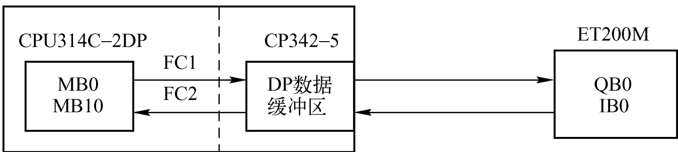



图 8-131　主站与从站之间的数据交换

在 OB1 中，编写程序如图 8-132 所示。

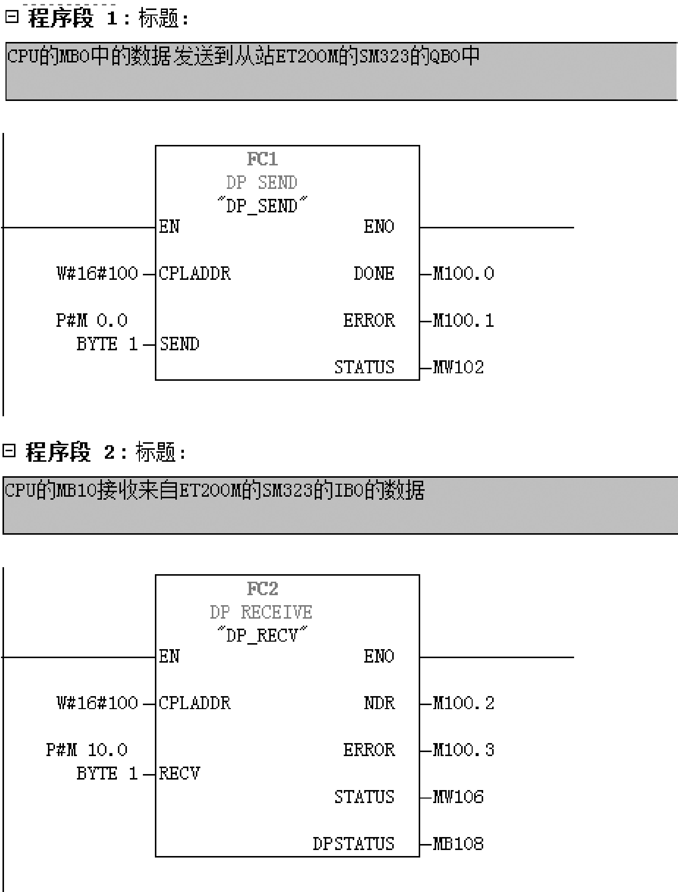



图 8-132　OB1 中的程序

### 8.4.7　CP342 -5 作从站的 PROFIBUS - DP 通信

以下介绍 CP342 -5 作从站时，与 CUP412 -1 之间的 PROFIBUS 通信的实例。

**【例 8-9】**有 1 台设备，控制系统由 CUP412 -1、CPU 314C -2DP、CP342 -5、SM421

和 SM422 组成，要求实时从 CPU314C－2DP 的 MW0 发出 2 个字节到主站的 CUP412－1 的 MW0，从主站的 CUP412－1 的 MW10 发送 2 个字节到 CPU314C－2DP 的 MW10 中。

**解：**

（1）主要软硬件配置

① 1 套 STEP 7 V5.5 SP4。

② 1 台 CPU 314C－2DP。

③ 1 台 CP342－5。

④ 1 台 CUP412－1。

⑤ 1 台 SM421 和 SM422。

⑥ 1 根编程电缆。

⑦ 1 根 PROFIBUS 网络电缆（含两个网络总线连接器）。

PROFIBUS 现场总线硬件配置图如图 8-133 所示。

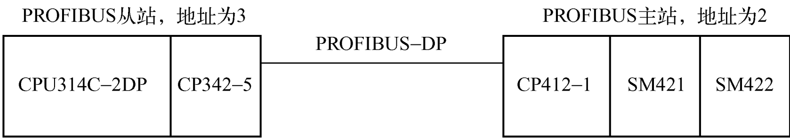



图 8-133　PROFIBUS 现场总线硬件配置图

（2）硬件组态

1）新建项目并组态从站。在 SIMATIC Manager 界面，首先新建一个项目，本例为“CP_SLAVE”，再插入 S7－300 站点；打开硬件组态界面，插入机架，再插入 CPU 314C－2DP 和 CP342－5，如图 8-134 所示，双击“CP342－5”，弹出如图 8-135 所示的“属性－CP342－5”界面，单击“属性”按钮，弹出“属性－PROFIBUS 接口”界面，单击“新建”按钮，选择站地址为“3”，单击“确定”按钮。




图 8-134　从站的硬件组态




图 8-135　新建 PROFIBUS 网络

回到“属性 - CP342 - 5”界面，选择“工作模式”选项卡，选定“DP 从站”模式，再单击“确定”按钮，如图 8-136 所示，单击工具栏的“保存和编译”按钮，如没有显示错误，从站硬件组态工作完成。




图 8-136　DP 从站模式

2）组态主站。在 SIMATIC Manager 界面插入 S7 - 400 站点；打开硬件组态界面，插入机架 UR2，再先后插入 PS407 4A、CPU 412 - 1、SM421 和 SM422，如图 8-137 所示。

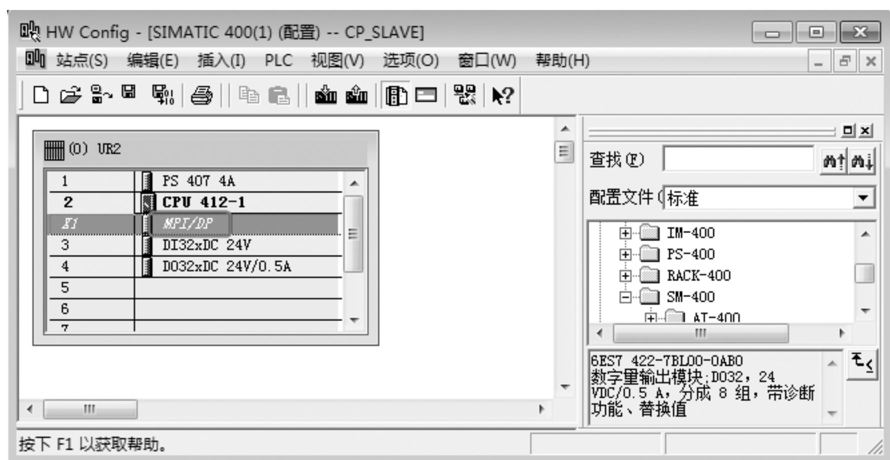



图 8-137 硬件组态

3）新建 PROFIBUS 网络。在图 8-137 中，双击“MPI/DP”弹出“属性 - MPI/DP”对话框，把类型选为“PROFIBUS”，单击“属性”按钮，弹出“属性 - PROFIBUS 接口”对话框，单击“新建”按钮，实际就是新建 PROFIBUS 网络，如图 8-138 所示，单击“属性 - PROFIBUS 接口”对话框的“确定”按钮，最后单击“属性 - MPI/DP”对话框中的“确定”按钮。




图 8-138 新建 PROFIBUS 网络

先选中如图 8-139 所示的 PROFIBUS 网线（标记为“1”处），选定硬件组态的目录树下“PROFIBUS DP”→“Configured Station”→“S7 300 CP342 - 5 DP”→“6GK6 342 - 5DA02 - 0XE0”→“V6.0”，按住鼠标左键，将其拖入如图 8-139 所示的网线位置，再松开鼠标。此时弹出“属性 - DP 从站”对话框，如图 8-140 所示，单击“连接”按钮。

选中“CP342 - 5”，分别将“2 bytes DI/Total Consistency”和“2 bytes DO/Total Consistency”拖入从站的 1 号和 2 号槽。单击工具栏的“保存和编译”按钮，如没有显示错误，

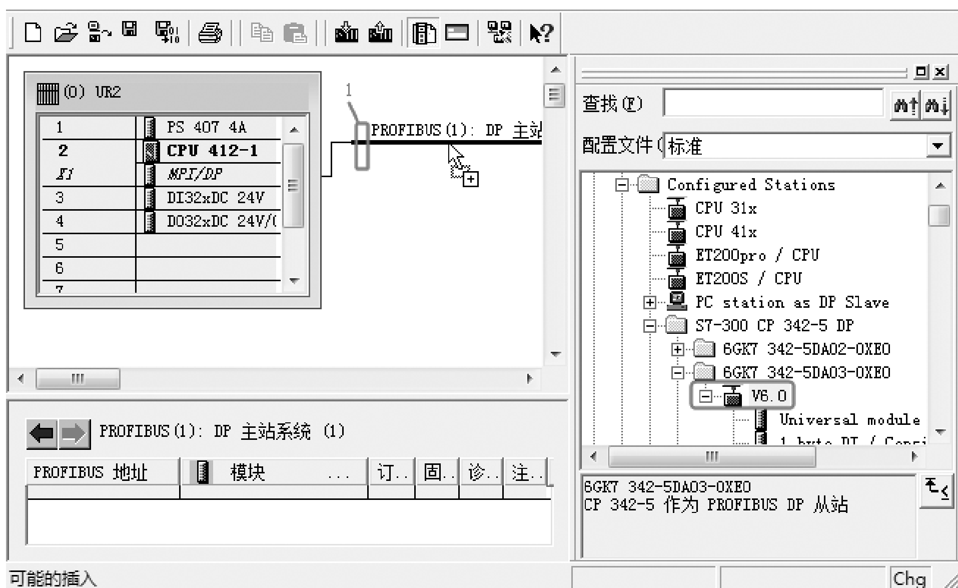



图 8-139　组态从站

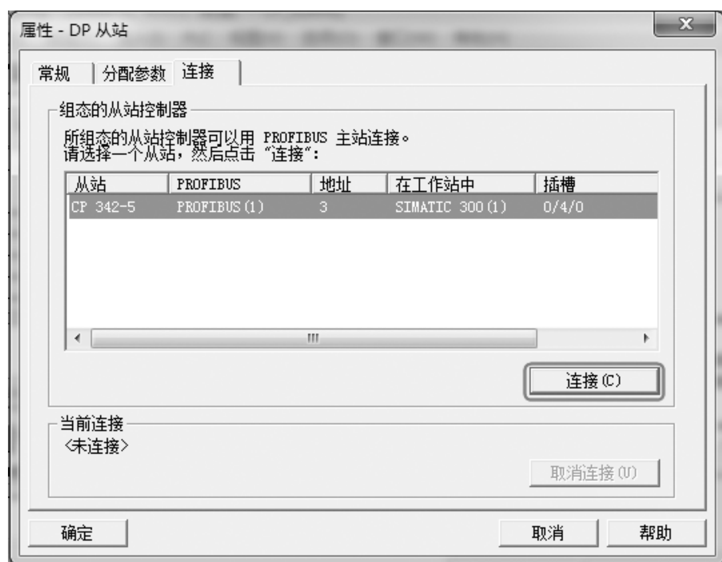



图 8-140　从站连接

硬件组态工作完成，如图 8-141 所示。

4）编写程序。与 CPU 集成的 DP 接口不同，CP342-5 作从站时，不能通过 I 和 Q 区直接交换数据到主站 CPU412-1 中，需要调用 FC1（DP_SEND）和 FC2（DP_RECV），建立虚拟通信口区访问主站。

从图 8-141 可以看到，主站输入数据区为 IB0～IB1，而输出数据区为 QB0～QB1，本例的主站与从站之间的数据交换如图 8-142 所示。

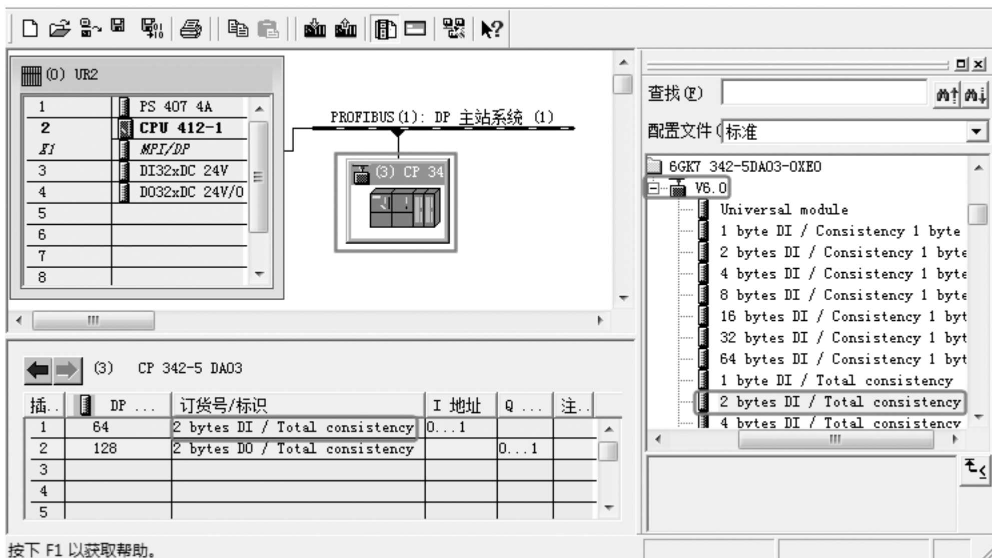



图 8-141　组态

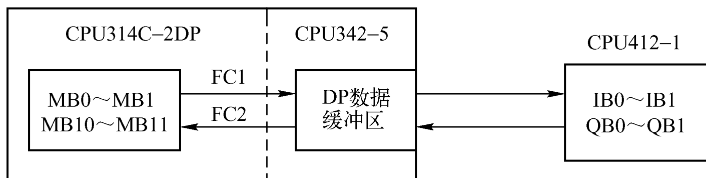
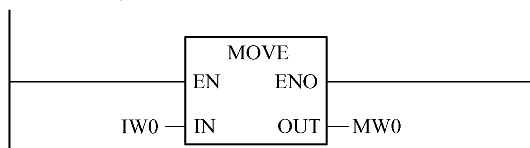



图 8-142　主站与从站之间的数据交换

主站的 OB1 中的程序如图 8-143 所示，从站的 OB1 中程序如图 8-144 所示。

程序段1：标题：




程序段2：标题：

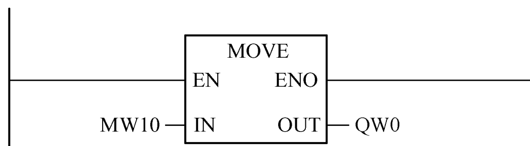
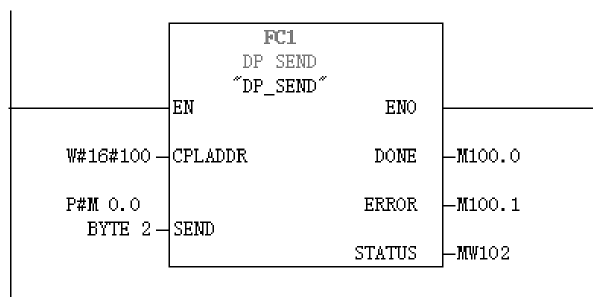



图 8-143　主站 OB1 中的程序

⊟ **程序段 1**：标题：

从站的MB0~MB1的数据发送到主站的IB0~IB1




⊟ **程序段 2**：标题：

从站MB10~MB11接收来自主站QB0~QB1的数据

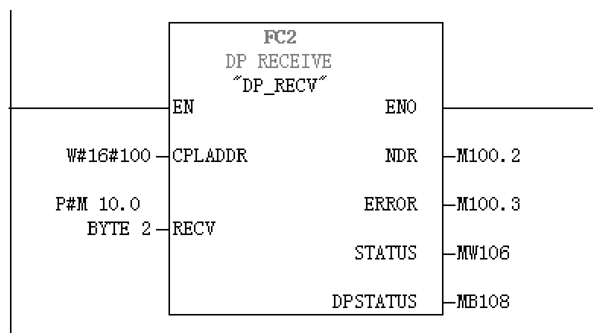



图 8-144　从站 OB1 中的程序

## 8.4.8　DP/DP Coupler 的 PROFIBUS－DP 通信

### 1. DP/DP Coupler 模块概述

DP/DP Coupler 用于连接两个 PROFIBUS－DP 主站网络，以便在这两个主站网络之间进行数据通信，数据通信区最高可以达 244 B 的输入和 244 B 的输出。DP/DP Coupler 模块面板如图 8-145 所示。

对于 DP/DP Coupler 连接的两个网段，通信速率可以不同，因此 DP/DP Coupler 非常适用于不同通信速率的两个 PROFIBUS－DP 主站系统之间的数据通信，但是对于通信数据区，网络 1 的输入区必须和网络 2 的输出区完全对应，同样网络 2 的输入区必须和网络 1 的输出区完全对应，否则会造成通信故障。

DP/DP Coupler 的拨码开关和指示灯的含义见表 8-11。

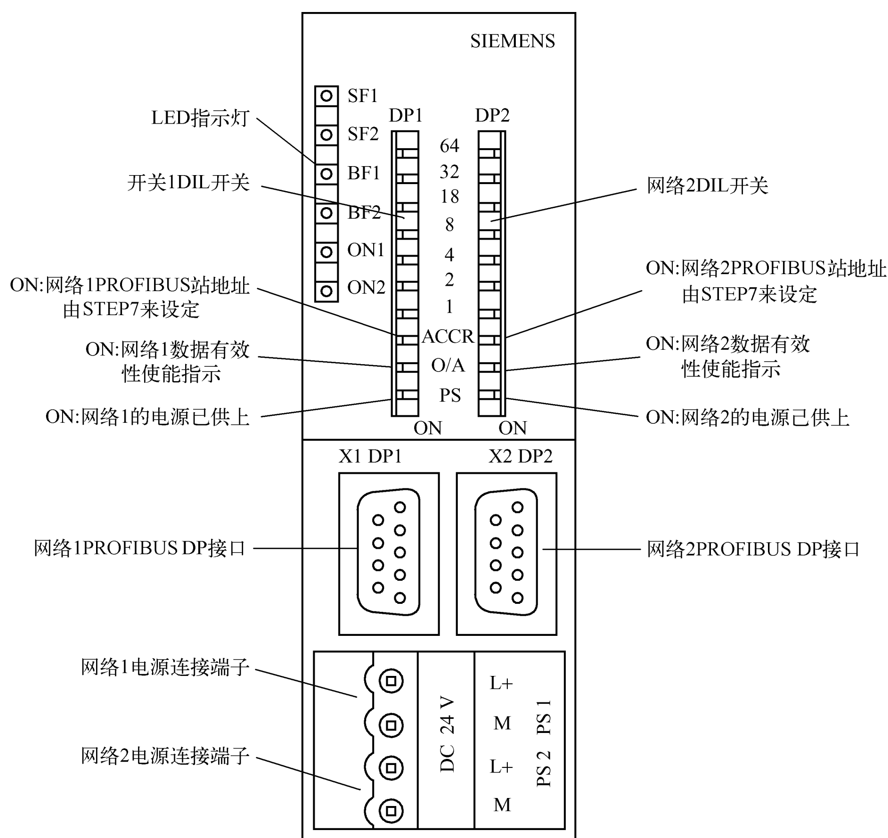



图 8-145　DP/DP Coupler 面板的外形

**表 8-11　拨码开关和指示灯的含义**

| 开　　关 | 拨　码　值 | | 含　　义 |
|---|---|---|---|
| PS | DP1 | ON | PS1 DC 24 V 供电监控未使能（用于诊断） |
| | | OFF | PS1 DC 24 V 供电监控未使能 |
| | DP2 | ON | PS2 DC 24 V 供电监控使能（用于诊断） |
| | | OFF | PS2 DC 24 V 供电监控未使能 |
| DIA | DP1 | ON | 网络 2 的输出数据发送给网络 1 的输入数据验证使能 |
| | | OFF | 网络 2 的输出数据发送给网络 1 的输入数据验证未使能 |
| | DP2 | ON | 网络 1 的输出数据发送给网络 2 的输入数据验证使能 |
| | | OFF | 网络 1 的输出数据发送给网络 2 的输入数据验证未使能 |
| ADDR | DP1 | ON | 网络 1PROFIBUS 站地址通过 STEP 7 来设置 |
| | | OFF | 网络 1PROFIBUS 站地址通过模块本身 DIL 开关来设置 |
| | DP2 | ON | 网络 2PROFIBUS 站地址通过 STEP 7 来设置 |
| | | OFF | 网络 2PROFIBUS 站地址通过模块本身 DIL 开关来设置 |
| 1，2，4，8，16，32，64 | DP1 | | 网络 1PROFIBUS 站地址设置开关（1～125） |
| | DP2 | | 网络 2PROFIBUS 站地址设置开关（1～125） |

**2. DP/DP Coupler 模块具有的特点**

1）连接两个不同的 PROFIBUS 网络进行通信，2 个网络的通信速率，站地址可以不同。

2）最多可以建立 16 个 I/O 数据交换区。

3）两个网络电气隔离，一个网段故障不影响另一个网段的运行。

4）支持 DPV1 全模式诊断。

5）可通过 DIL 开关，STEP 7 或其他编程工具设定 PROFIBUS 站地址。

6）双路冗余供电方式。

**3. DP/DP Coupler 模块应用举例**

以下用一个例子，介绍 DP/DP Coupler 模块的应用，通过组建两个 PROFIBUS 网络，实现两个网络主站之间的 PROFIBUS 通信。

**【例 8-10】** 有 1 台设备，控制系统由 CUP412－1、CPU 314C－2DP、DP/DP Coupler、CPU226CN、EM277、SM421 和 SM422 组成，要求实时从 CPU314C－2DP 的 MW0 发出 2 个字节到 CUP412－1 的 MW0，从 CUP412－1 的 MW10 发送 2 个字节到 CPU314C－2DP 的 MW10 中。

**解：**

（1）主要软硬件配置

① 1 套 STEP 7 V5.5 SP4。

② 1 台 CPU 314C－2DP、CPU226CN 和 EM277。

③ 1 台 DP/DP Coupler。

④ 1 台 CUP412－1。

⑤ 1 台 CPU412－1、PS407、SM421 和 SM422。

⑥ 1 根编程电缆。

⑦ 两根 PROFIBUS 网络电缆（含 6 个网络总线连接器）。

PROFIBUS 现场总线硬件配置图如图 8-146 所示。

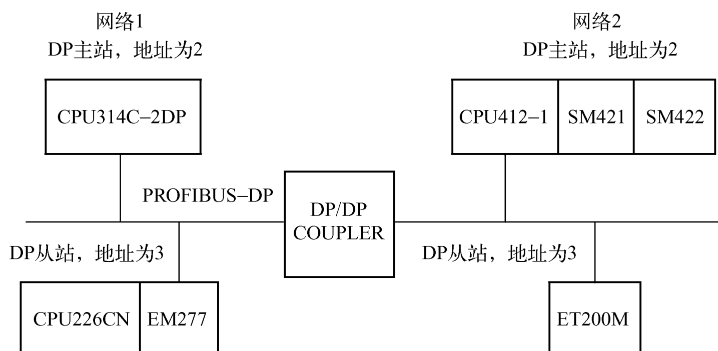



图 8-146　PROFIBUS 现场总线硬件配置图

（2）硬件组态

1）新建项目，并组态网络 1。在 SIMATIC Manager 界面新建一个项目，本例为“DP/DP Coupler”，再插入 S7－300 站点；打开硬件组态界面，插入机架，再插入 CPU 314C－

2DP，如图 8-147 所示，双击“DP”，弹出“属性”界面，单击“属性”按钮，弹出“属性-PROFIBUS 接口”界面，单击“新建”按钮，选择站地址为“2”，单击“确定”按钮。再插入 EM277 模块，EM277 的 DP 地址为“3”，然后插入输入/输出区间。

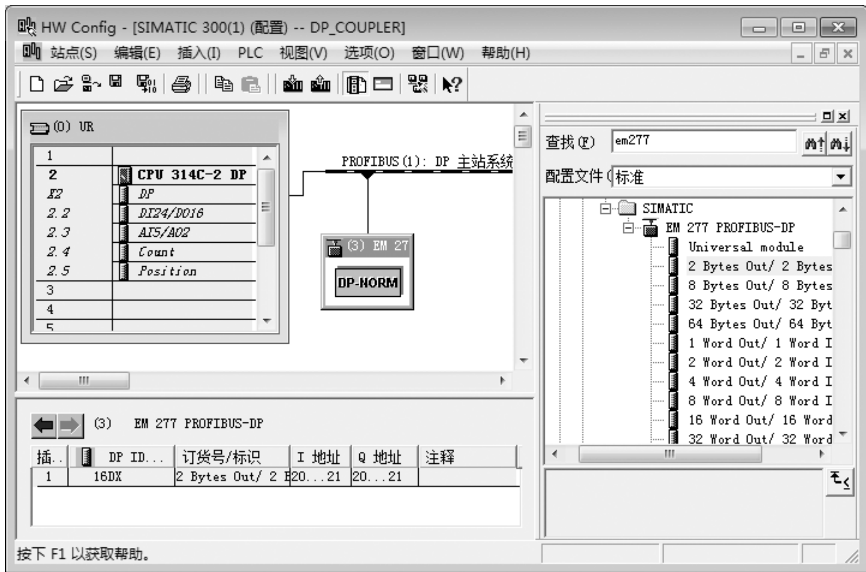


图 8-147　网络 1 的硬件组态

2）插入“DP/DP Coupler”模块。用鼠标左键选中“DP/DP Coupler”，并拖入到如图 8-148 所示的箭头所示的位置，弹出“属性-PROFIBUS 接口”界面，如图 8-149 所示，选择地址为“4”，再单击“确定”按钮。

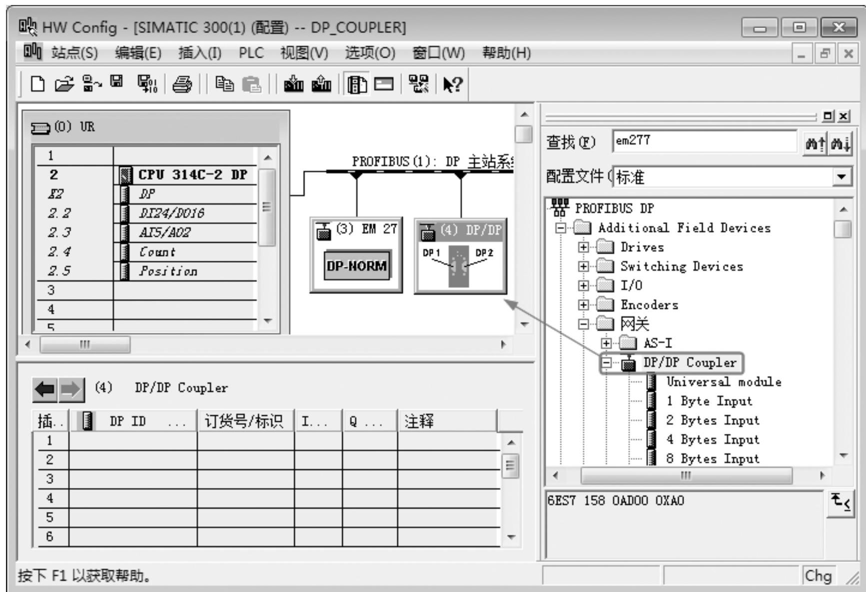


图 8-148　插入 DP/DP Coupler




图 8-149　新建 PROFIBUS 网络

再插入输入/输出区间，用鼠标左键选中“2 Byte Input”并拖入如图 8-150 所示的箭头所示的位置，用同样的方法对“2 Byte Output”进行操作。单击工具栏的“保存和编译”按钮，如没显示错误，网络 1 的硬件组态工作完成。

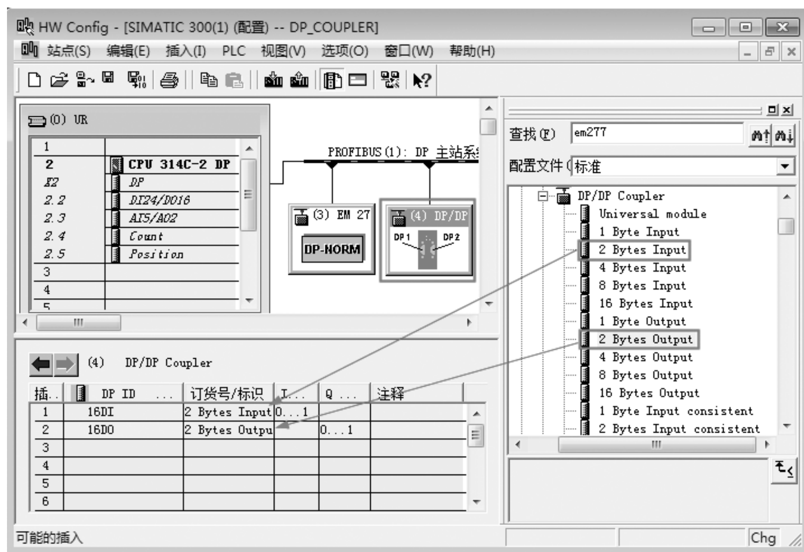



图 8-150　插入输入/输出区间

3）组态网络 2。在 SIMATIC Manager 界面，插入 S7－400 站点；打开硬件组态界面，插入机架 UR2，再先后插入 PS407 4A、CPU 412－1、SM421 和 SM422，如图 8-151 所示。

4）新建 PROFIBUS 网络。双击“MPI/DP”，如图 8-151 所示，弹出“属性－MPI/DP”对话框，把类型选为“PROFIBUS”，再单击“属性”按钮，弹出“属性－PROFIBUS 接口”对话框，单击“新建”按钮，实际就是新建 PROFIBUS 网络，如图 8-152 所示，单击“属性－PROFIBUS 接口”对话框的“确定”按钮，最后单击“属性－MPI/DP”对话框的“确定”按钮。

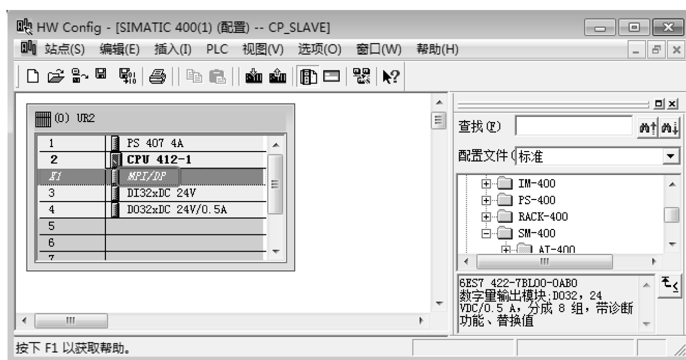



图 8-151　硬件组态




图 8-152　新建 PROFIBUS 网络

5）插入“DP/DP Coupler”模块。用鼠标左键选中“DP/DP Coupler”，并拖入图 8-153 中，弹出“属性 - PROFIBUS 接口”界面，如图 8-154 所示，选择地址为“5”，再单击“确定”按钮。

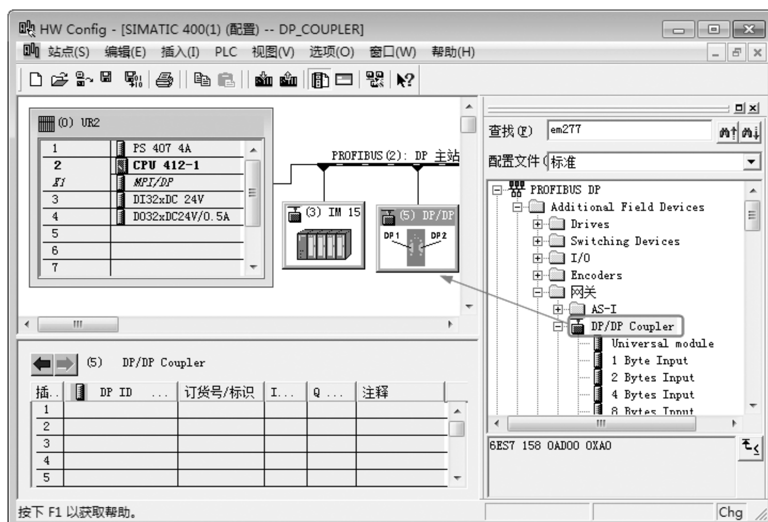



图 8-153　拖入 DP/DP Coupler




图 8-154 新建 PROFIBUS 网络

再插入输入/输出区间，用鼠标左键选中“2 Byte Input”，并拖入如图 8-155 所示的箭头所示的位置，用同样的方法对，“2 Byte Output”进行操作。单击工具栏的“保存和编译”按钮，如没有显示错误，网络 2 的硬件组态工作完成。打开网络组态如图 8-156 所示。

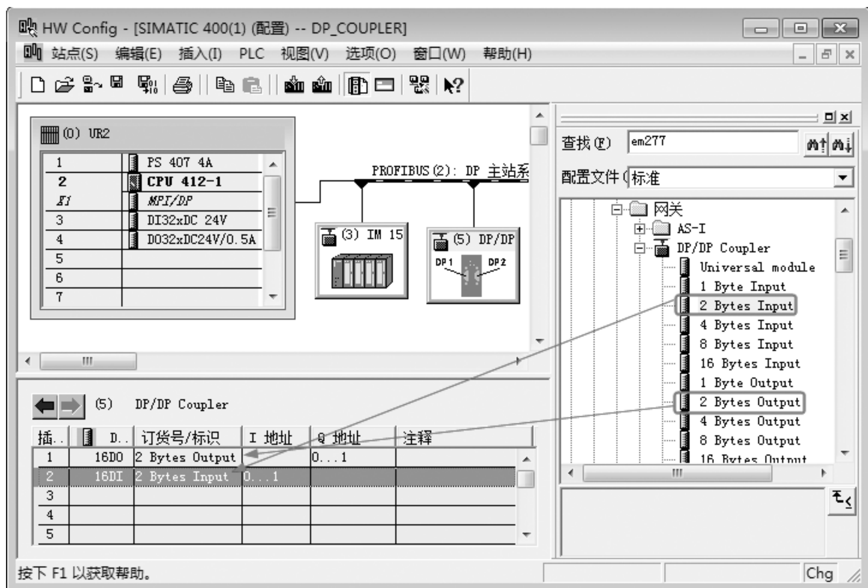



图 8-155 插入输入/输出区间

（3）编写程序

网络 1 的 OB1 中的程序如图 8-157 所示，网络 2 的 OB1 中的程序如图 8-158 所示。

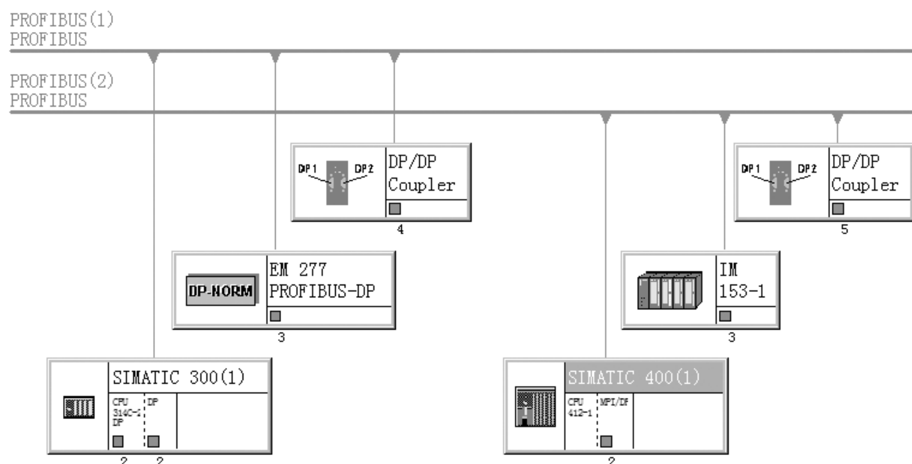
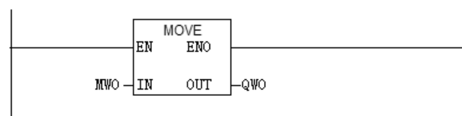



图 8-156　网络组态

⊟ **程序段 1**：标题：

MW0的数据传送到网络1主站QW0，网络1主站QW0自动发送到网络2主站的IW0




⊟ **程序段 2**：标题：

网络1主站IW0自动接收网络2的QW0发送的数据，网络1主站IW0把数据传送到MW10

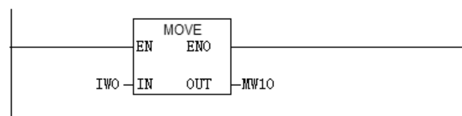
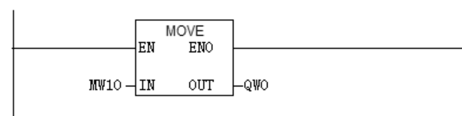



图 8-157　网络 1 的 OB1 中的程序

⊟ **程序段 1**：标题：

MW10的数据传送到网络2主站QW0，网络2主站QW0自动发送到网络1主站的IW0




⊟ **程序段 2**：标题：

网络2主站IW0自动接收网络1的QW0发送的数据，网络2主站IW0把数据传送到MW0

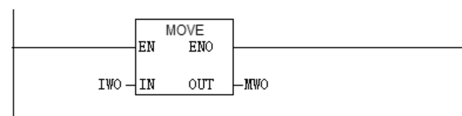



图 8-158　网络 2 的 OB1 中的程序

## 8.5 以太网通信及其应用

以太网（Ethernet）指的是由 Xerox 公司创建、并由 Xerox、Intel 和 DEC 公司联合开发的基带局域网规范。以太网使用 CSMA/CD（载波监听多路访问及冲突检测技术）技术，并以 10 Mbit/s 的速率运行在多种类型的电缆上。以太网与 IEEE802.3 系列标准相类似。以太网不是一种具体的网络，而是一种技术规范。

### 8.5.1 以太网通信基础

**1. 以太网的历史**

以太网的核心思想是：使用公共传输信道。这个思想产生于 1968 年美国的夏威尔大学。

以太网技术的最初进展源自于施乐帕洛阿尔托研究中心的许多先锋技术项目中的一个。人们通常认为以太网发明于 1973 年，以当年罗伯特·梅特卡夫（Robert Metcalfe）给他的老板 PARC 写了一篇有关以太网潜力的备忘录为标志。

1979 年，梅特卡夫成立了 3Com 公司。3Com 联合迪吉多、英特尔和施乐共同将网络进行标准化、规范化。这个通用的以太网标准于 1980 年 9 月 30 日出台。

**2. 以太网的分类**

以太网分为标准以太网、快速以太网、千兆以太网和万兆以太网。

**3. 以太网的连接**

（1）拓扑结构

1）星形。管理方便、容易扩展、需要专用的网络设备作为网络的核心节点、需要更多的网线且对核心设备的可靠性要求高。采用专用的网络设备（如集线器或交换机）作为核心节点，通过双绞线将局域网中的各台主机连接到核心节点上，这就形成了星形结构。星形网络虽然需要的线缆比总线型多，但布线和连接器比总线型的要便宜。此外，星形拓扑可以通过级联的方式很方便地将网络扩展到很大的规模，因此得到了广泛的应用，被绝大部分的以太网所采用。如图 8-159 所示，1 台 ESM（Electrical Switch Module）交换机与两台 PLC 和两台计算机组成星形网络，这种拓扑结构在工控中很常见。

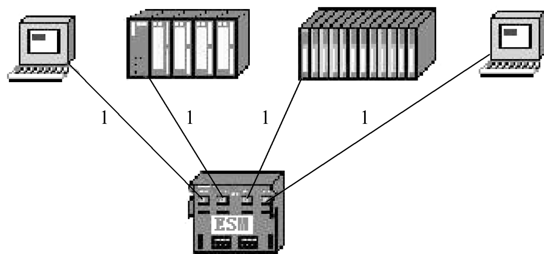



图 8-159　拓扑图

1—TP 电缆，RJ45 接口

2）总线型。如图 8-160 所示，所需的电缆较少、价格便宜、管理成本高、不易隔离故障点、采用共享的访问机制、易造成网络拥塞。早期以太网多使用总线型的拓扑结构，采用同轴缆作为传输介质，连接简单，通常在小规模的网络中不需要专用的网络设备，但由于它存在的固有缺陷，已经逐渐被以集线器和交换机为核心的星形网络所代替。如图 8-160 所示，3 台交换机组成总线网络，交换机再与 PLC、计算机和远程 IO 模块组成网络。

3）环形。如图 8-161 所示。西门子的网络中，用 OLM（Optical Link Module）模块将网络首位相连，形成环网，也可用 OSM（Optical Switch Module）交换机组成环网。与总线型相比冗余环网增加了交换数据的可靠性。如图 8-161 所示，4 台交换机组成环网，交换机

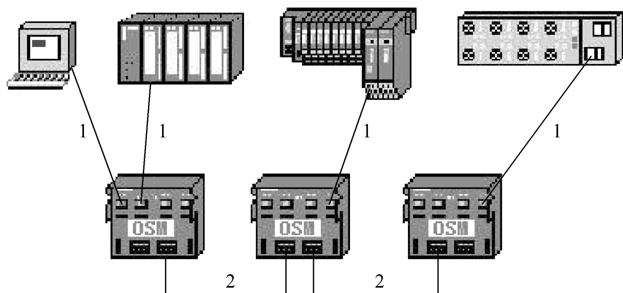



图 8-160　总线型拓扑应用
1—TP 电缆，RJ45 接口　2—光缆

再与 PLC、计算机和远程 IO 模块组成网络，这种拓扑结构在工控中也很常见。

此外，还有网状和蜂窝状等拓扑结构。

（2）接口的工作模式

以太网卡可以工作在两种模式下：半双工和全双工。

（3）传输介质

以太网可以采用多种连接介质，包括同轴缆、双绞线和光纤等。其中双绞线多用于从主机到集线器或交换机的连接，而光纤则主要用于交换机间的级联和交换机到路由器间的点到点链路上。同轴缆作为早期的主要连接介质已经逐渐趋于淘汰。

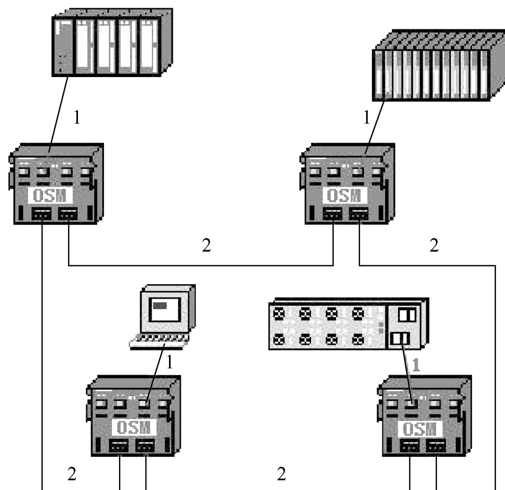



图 8-161　环形拓扑应用
1—TP 电缆，RJ45 接口　2—光缆

总之，以太网是目前世界上最为流行的拓扑标准之一，具有传播速度高、网络资源丰富、系统功能强大、安装简单和使用维修方便等优点。

**4. 工业以太网通信简介**

（1）初识工业以太网

所谓工业以太网，通俗地讲就是应用于工业的以太网，是指在技术上与商用以太网（IEEE802.3 标准）兼容，但材质的选用、产品的强度和适用性方面应能满足工业现场的需要。工业以太网技术的优点表现在：以太网技术应用广泛，为所有的编程语言所支持；软硬件资源丰富；易于与 Internet 连接，实现办公自动化网络与工业控制网络的无缝连接；通信速度快；可持续发展的空间大等。

为促进以太网在工业领域的应用，国际上成立了工业以太网协会（Industrial Ethernet Association，IEA）。

（2）网络电缆接法

用于以太网的双绞线有 8 芯和 4 芯两种，双绞线的电缆连线方式也有两种，即正线（标准 568B）和反线（标准 568A），其中正线也称为直通线，反线也称为交叉线。正线接线如图 8-162 所示，两端线序一样，从上至下线序是：白绿、绿、白橙、蓝、白蓝、橙、白棕

和棕。反线接线如图 8-163 所示，一端为正线的线序，另一端为反线线序，从上至下线序是：白橙、橙、白绿、蓝、白蓝、绿、白棕和棕。也就是 568B 标准。对于千兆以太网，用 8 芯双绞线，但接法不同于以上所述的接法，请参考有关文献。

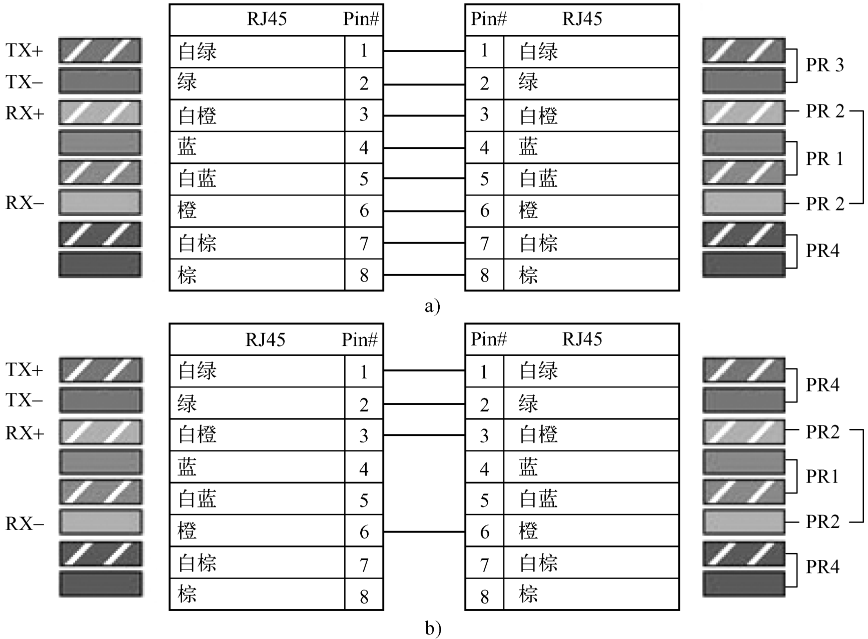



图 8-162　双绞线正线接线图

a）8 芯线　b）4 芯线

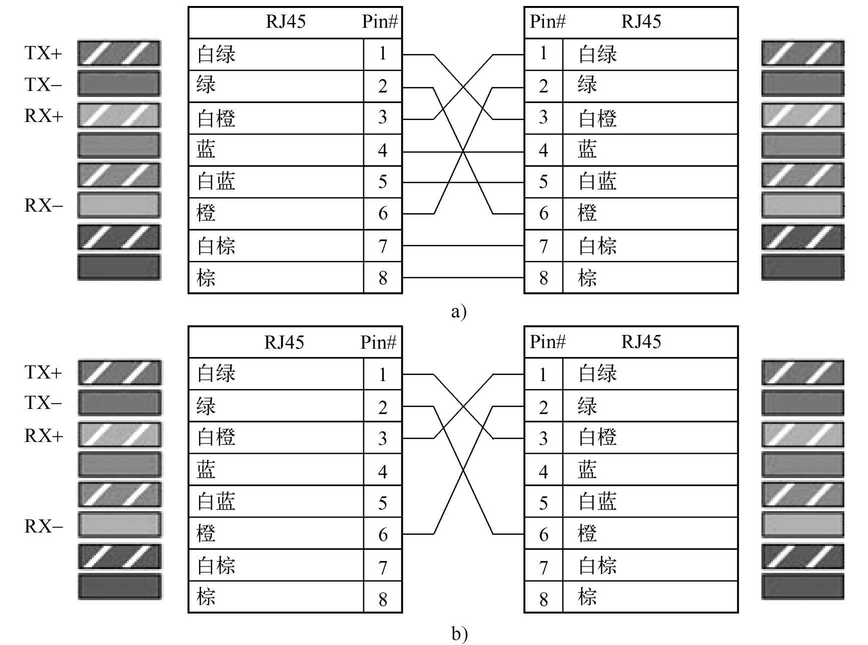



图 8-163　双绞线反线接线图

a）8 芯线　b）4 芯线

对于 4 芯的双绞线，只用连接头上的（常称为水晶接头）1、2、3 和 6 四个引脚。西门子的 PROFINET 工业以太网采用 4 芯的双绞线。

### 8.5.2　S7 – 300 间的以太网通信

**1. 西门子工业以太网通信方式简介**

工业以太网的通信主要利用第 2 层（ISO）和第 4 层（TCP）的协议。以下是西门子以太网的几种通信方式。

（1）ISOTransport（ISO 传输协议）

ISO 传输协议支持基于 ISO 的发送和接收，使得设备（例如 SIMATIC S5 或 PC）在工业以太网上的通信非常容易，该服务支持大数据量的数据传输（最大 8 KB）。ISO 数据接收由通信方确认，通过功能块可以看到确认信息。用于 SIMATIC S5 和 SIMATIC S7 的工业以太网连接。

（2）ISO – on – TCP

ISO – on – TCP 支持第 4 层 TCP/IP 协议的开放数据通信。用于支持 SIMATIC S7 和 PC 以及非西门子支持的 TCP/IP 以太网系统。ISO – on – TCP 符合 TCP/IP，但相对于标准的 TCP/IP 还附加了 RFC 1006 协议，RFC 1006 是一个标准协议，该协议描述了如何将 ISO 映射到 TCP 上去。

（3）UDP

UDP（User Datagram Protocol，用户数据报协议），属于第 4 层协议，提供了 S5 兼容通信协议，适用于简单的交叉网络数据传输，没有数据确认报文，不检测数据传输的正确性。UDP 支持基于 UDP 的发送和接收，使得设备（例如 PC 或非西门子公司设备）在工业以太网上的通信非常容易。该协议支持较大数据量的数据传输（最大 2 KB），数据可以通过工业以太网或 TCP/IP 网络（拨号网络或因特网）传输。通过 UDP，SIMATIC S7 通过建立 UDP 连接，提供了发送/接收通信功能，与 TCP 不同，UDP 实际上并没有在通信双方建立一个固定的连接。

（4）TCP/IP

TCP/IP 中传输控制协议，支持第 4 层 TCP/IP 协议的开放数据通信，提供了数据流通信，但并不将数据封装成消息块，因而用户并不接收到每一个任务的确认信号。TCP 支持面向 TCP/IP 的 Socket。

TCP 支持给予 TCP/IP 的发送和接收，使得设备（例如 PC 或非西门子设备）在工业以太网上的通信非常容易。该协议支持大数据量的数据传输（最大 8 KB），数据可以通过工业以太网或 TCP/IP 网络（拨号网络或因特网）传输。通过 TCP，SIMATIC S7 可以通过建立 TCP 连接来发送/接收数据。

**2. S7 通信**

S7 通信（S7 Communication）集成在每一个 SIMATIC S7/M7 和 C7 的系统中，属于 OSI 参考模型第 7 层应用层的协议，它独立于各个网络，可以应用于多种网络（MPI、PROFIBUS 及工业以太网）。S7 通信通过不断地重复接收数据来保证网络报文的正确。在 SIMATIC S7 中，通过组态建立 S7 连接来实现 S7 通信。在 PC 上，S7 通信需要通过 SAPI – S7 接口函数或 OPC（过程控制用对象链接与嵌入）来实现。

**3. 实例**

以下用两台 S7 -300 的以太网通信为例，介绍 S7 -300 间的以太网通信。

**【例 8-11】** 有两台设备，分别由一台 CPU 314C -2DP 控制，要求从设备 1 上的 CPU314C -2DP 的 MB10 发出 1 个字节到设备 2 的 CPU314C -2DP 的 MB10，从设备 2 上的 CPU314C -2DP 的 MB20 发出 1 个字节到设备 1 的 CPU314C -2DP 的 MB20。

**解：** S7 -300 之间的组态可以采用很多连接方式，如 TCP、ISO - on - TCP 和 S7 Communication 等，以下仅介绍 TCP 连接方式。

（1）软硬件配置

S7 -300 间的以太网通信硬件配置如图 8-164 所示，本例用到的软硬件如下：

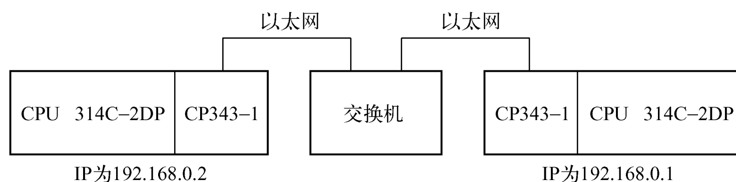



图 8-164　S7 -300 间的以太网通信硬件配置图

① 两台 CPU 314C -2DP。
② 两台 CP343 -1 以太网模块。
③ 1 根 PC/MPI 适配器（USB 口）。
④ 1 台个人电脑（含网卡）。
⑤ 1 台 8 口交换机。
⑥ 两根带 RJ45 接头的 8 芯双绞线（正线）。
⑦ 1 套 STEP 7 V5. 5 SP4。

（2）硬件组态

1）新建项目。新建项目，命名为“Enet_ TCP”，再插入两个站分别是 CLIENT 和 SERVER，每个站点上，配置一台 CP343 -1 以太网通信模块，如图 8-165 所示。

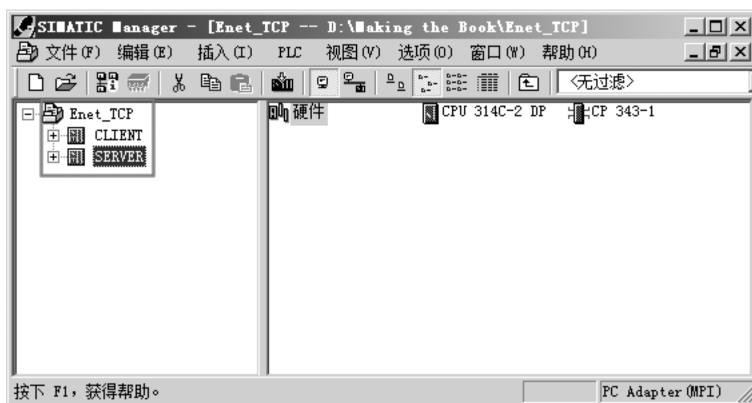



图 8-165　新建项目

【关键点】西门子工业以太网通信中，客户端（CLIENT）是主控站，服务器端（SERVER）是被控站。

2）组态以太网。双击“硬件”，弹出如图 8-166 所示界面，选中“CP 343-1”的“PN-IO”，并双击之，弹出如图 8-167 所示界面，单击“属性”按钮，弹出如图 8-168 所示界面。




图 8-166 组态以太网（1）

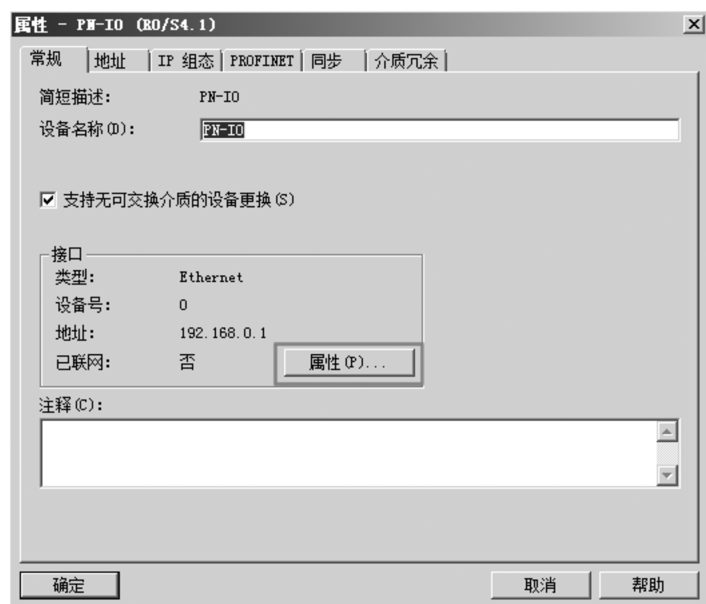



图 8-167 组态以太网（2）

3）新建网络。单击“新建”按钮，弹出如图 8-168 所示界面，单击“确定”按钮，弹出如图 8-169 所示界面，再单击“确定”按钮，弹出如图 8-170 界面。




图 8-168　新建以太网（1）




图 8-169　新建以太网（2）

4）设置网络参数。如图 8-170 所示，先选中“Ethernet（1）”，再在“IP 地址”中设置为：“192.168.0.1”，在“子网掩码”中设置为：“255.255.255.0”，单击“确定”按钮。




图 8-170　设置网络参数

5）采用同样的方法，配置第二个以太网模块的参数，不同之处在于，将“IP 地址”设置成“192.168.0.2”。

**【关键点】** 同一个网络中，IP 地址是唯一的，绝对不允许重复。

6）打开网络连接。在 SIMATIC 管理界面中，如图 8－171 所示，先选中“Ethernet（1）”，再双击“Ethernet（1）”，弹出如图 8–172 所示界面。

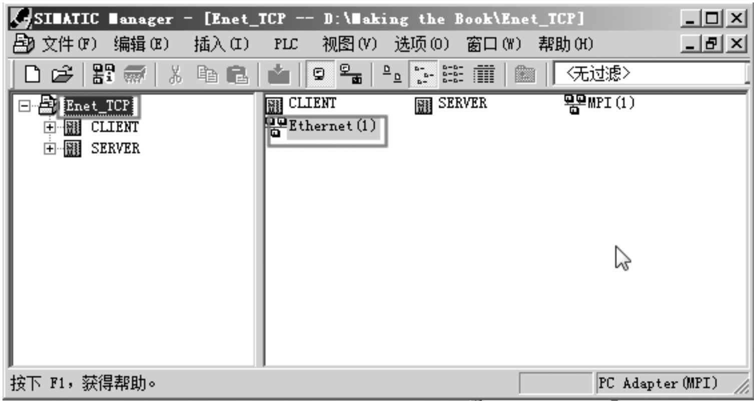



图 8–171　打开网络连接界面

7）组态以太网连接。如图 8–172 所示，先选中客户端的“1”处，单击鼠标右键，弹出快捷菜单，然后单击“插入新连接”，弹出如图 8–173 所示界面。

**【关键点】** 若一个 PLC 中选择了“插入新连接”选项，另一 PLC 则不必激活此项，必须有一台 PLC 选择此选项，以便在通信初始化中起到主动连接的作用。

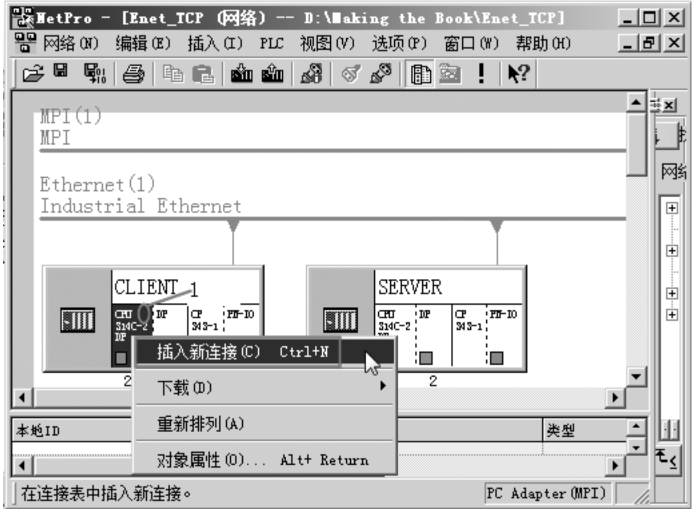



图 8–172　组态以太网连接

8）添加一个 TCP 连接。如图 8–173 所示，先选中“CPU 314C－2DP”，再选择“TCP 连接”，然后单击“应用”按钮，弹出如图 8–174 所示界面。

9）设置网络连接参数。如图 8–174 所示，先选择“激活连接”，再单击“确定”按钮。

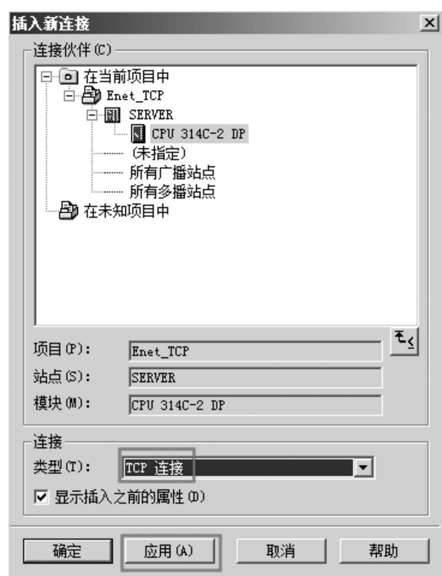



图 8-173　添加一个 TCP 连接




图 8-174　设置网络连接参数

在如图 8-175 中的“地址”选项卡中可以看到通信双方的 IP 地址，占用的端口号可以自行设置，也可以使用默认值，如 2001。编译后存盘，至此硬件组态完成。




图 8-175　设置 TCP/IP 端口

**【关键点】** 图 8-174 中的 ID 是组态时的连接号，LADDR 是模块硬件组态逻辑地址，地址相同才能通信，在编程时要用到。

（3）相关指令简介

AG_SEND 块将数据传送给以太网 CP，用于在一个已组态的 ISO 传输连接上进行传输。所选择的数据区可以是一个位存储器区或一个数据块区。当可以在以太网上发送整个用户数据区时，指示无错执行该功能。AG_SEND 的各项参数见表 8-12。

**表 8-12　AG_SEND（FC5）指令格式**

| LAD | 输入/输出 | 含　义 | 数 据 类 型 |
|---|---|---|---|
| "AG_SEND"<br>EN　ENO<br>ACT　DONE<br>ID　ERROR<br>LADDR　STATUS<br>SEND<br>LEN | EN | 使能 | BOOL |
| | ACT | 发送请求 | BOOL |
| | ID | 组态时的连接号 | INT |
| | LADDR | 模块硬件组态地址 | WORD |
| | SEND | 发送的数据区 | ANY |
| | LEN | 发送数据长度 | INT |
| | ERROR | 错误代码 | BOOL |
| | STATUS | 返回数值（如错误值） | WORD |
| | DONE | 发送是否完成 | BOOL |

AG_RECV 功能（FC）接收从以太网 CP 在已组态的连接上传送的数据。为数据接收指定的数据区可以是一个位存储区或一个数据块区。当可以从以太网 CP 上接收数据时，指示无错执行该功能。AG_RECV 的各项参数见表 8-13。

**表 8-13　AG_RECV 指令格式**

| LAD | 输入/输出 | 含　义 | 数 据 类 型 |
|---|---|---|---|
| "AG_RECV"<br>EN　ENO<br>ID　NDR<br>LADDR　ERROR<br>RECV　STATUS<br>LEN | EN | 使能 | BOOL |
| | ID | 组态时的连接号 | INT |
| | LADDR | 模块硬件组态地址 | WORD |
| | RECV | 接收数据区 | ANY |
| | NDR | 接收数据确认 | BOOL |
| | ERROR | 错误代码 | BOOL |
| | STATUS | 返回数值（如错误值） | WORD |
| | LEN | 接收数据长度 | INT |

（4）编写程序

在编写程序时，双方都要用到发送 AG_SEND（FC5）指令和接收 AG_RECV（FC6）指令，客户端（IP 地址为 192. 168. 0. 1）的梯形图如图 8-176 ~ 图 8-178 所示。

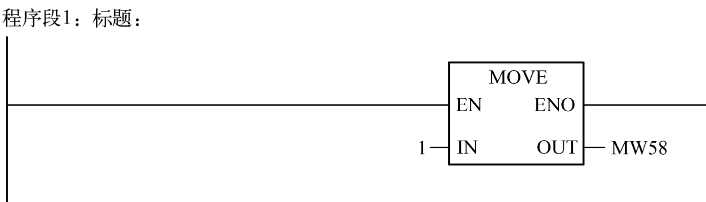



图 8-176　客户端 OB100 中的梯形图

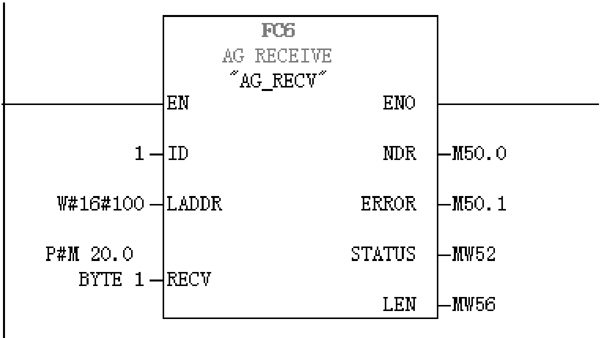



图 8-177　客户端 OB1 中的梯形图

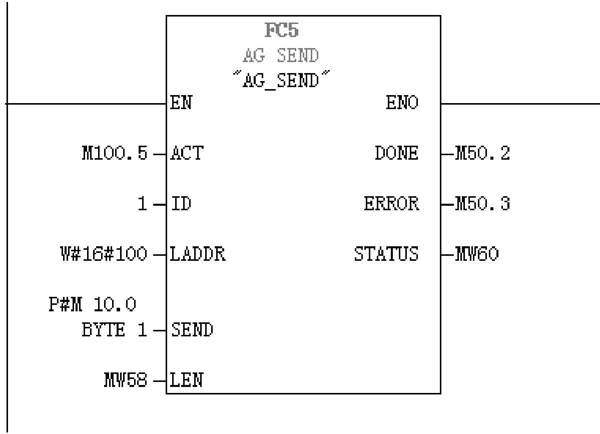



图 8-178　客户端 OB35 中的梯形图

服务器端（IP 地址为 192.168.0.2）中的梯形图程序如图 8-179 ~ 图 8-181 所示。

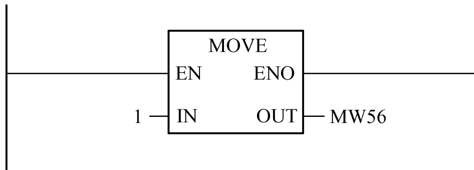



图 8-179　服务器端 OB100 中的梯形图

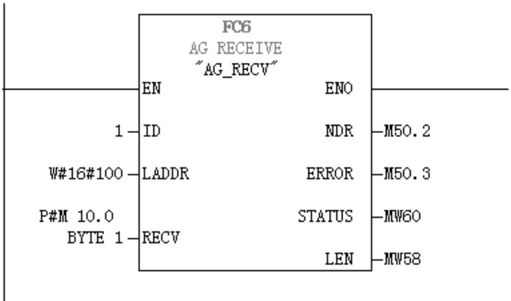



图 8-180　服务器端 OB1 中的梯形图

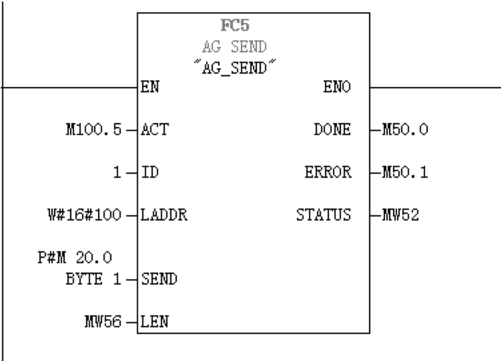



图 8-181　服务器端 OB35 中的梯形图

### 8.5.3　S7－400 与远程 IO 模块 ET200 间的 PROFINET 通信

**1. PROFINET 简介**

PROFINET 是 PROFIBUS 组织推出的基于工业以太网的开放式现场总线。PROFINET 自动化通信领域提供了一个完整的网络解决方案，囊括包括诸如实时以太网、运动控制、分布式自动化、故障安全及网络安全等自动化领域问题，并可以完全兼容工业以太网和现有的现场总线（如 PROFIBUS）技术。

PROFINET 可进行实时通信，如运动控制。PROFINET 目前是西门子主推的现场总线，已经取代 PROFIBUS 成为西门子公司的标准配置。

**2. 实例**

PROFINET 有两种应用形式，PROFINET IO 和 PROFINET CBA。PROFINET IO 适合于模块化分布式的应用，与 PROFIBUS－DP 方式相似。PROFINET CBA 适合于智能站点之间的应用。以下通过例子介绍 PROFINET IO 的应用。

**【例 8-12】**某系统的控制器由 S7－400、SM421、CP443－1、ET200M 和 SM323 组成，要用 S7－400 上的两个按钮控制远程站上的一台电动机的起停，请组态并编写相关程序。

**解：**

（1）主要软硬件配置

① 1 套 STEP 7 V5.5 SP4。

② 1 台 CP421－1。

③ 1 台 SM421 和 SM422。

④ 1 台 CP443－1。

⑤ 1 台 153－4 PN。

⑥ 1 台 EM323。

⑦ 1 根网线。

PROFINET 现场总线硬件配置如图 8-182 所示。




图 8-182　PROFINET 现场总线硬件配置

（2）硬件组态过程

1）新建项目。新建项目命名为“Profi_IO”，插入站点 CPU 400，双击“硬件”，打开硬件组态界面，如图 8-183 所示。

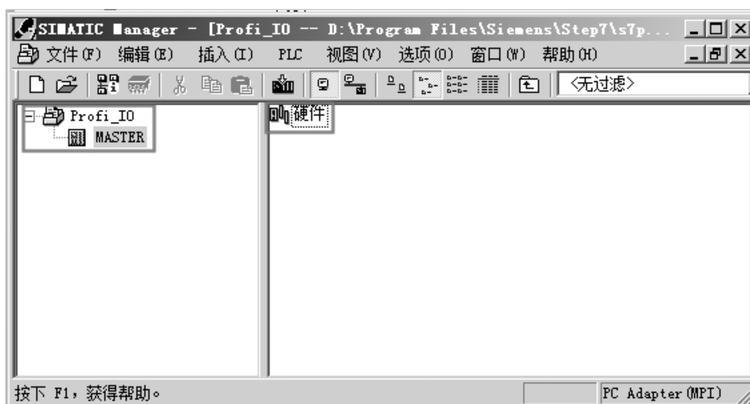


图 8-183　新建项目

2）组态硬件。先插入机架 UR2，再插入电源 PS 407 4A，然后插入 CP443 - 1、DO32 和 DI32 模块如图 8-184 所示。



图 8-184　组态硬件

3）设置主站 IP 地址。双击图 8-184 中的“PN - IO”，打开“PN - IO”的属性界面，单击“属性”按钮，如图 8-185 所示。弹出如图 8-186 所示的界面，按图中所示设置 IP 地

址，单击“确定”按钮即可。




图 8-185　PN－IO 属性




图 8-186　设置主站 IP 地址

4）网络组态。选中 PN－IO，右击鼠标，弹出快捷菜单，单击“插入 PROFINET IO 系统”，如图 8-187 所示。

选中标记“1”处，单击“PROFINET IO”→“I/O”→“ET 200M”→“IM153－4 PN”，如图 8-188 所示。然后在远程 IO 模块上插入信号模块 SM323，如图 8-189 所示。

5）分配设备名称和验证设备名称。PROFIBUS－DP 与远程 IO 通信的硬件组态做到上一步就完成了。因为远程 IO 的地址由拨码开关设置。而 IP 地址不能由拨码开关设置，因此把设备名称和 IP 地址绑定在一起，在验证设备时，将设备名称下载到远程 IO 模块的存储卡中，实际就是把 IP 地址下载到远程 IO 模块中。双击如图 8-189 所示的“IM153－4 PN”，弹出如图 8-190 所示界面，将设备“ET200M”与“192.168.0.2”关联，单击“确定”按钮，最后把整个项目用以太网下载到 CPU 中。



图 8-187　插入 PROFINET IO 系统

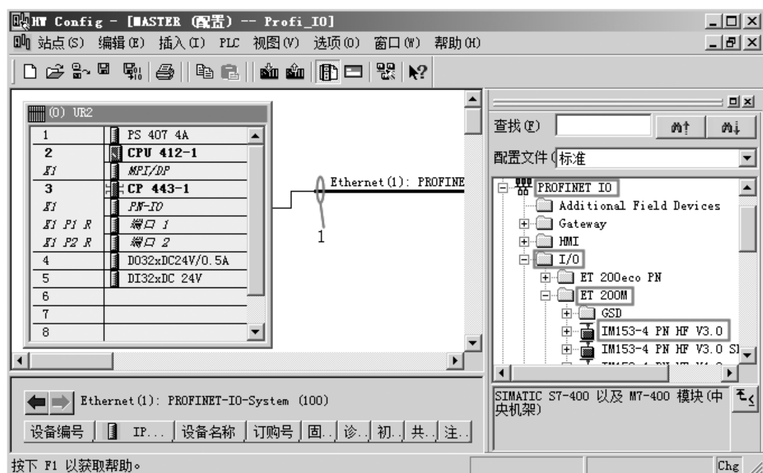


图 8-188　插入远程 IO 模块

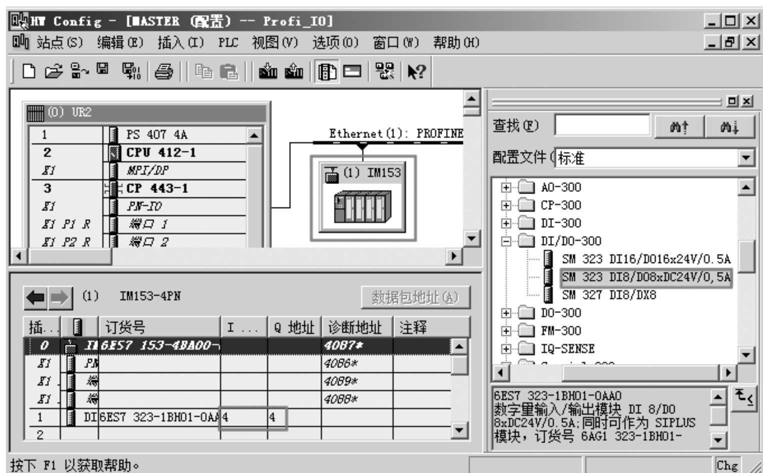


图 8-189　在远程 IO 模块上插入信号模块

图 8-190　设备重命名

在菜单栏中，单击“PLC”→“Ethernet”→“分配设备名称”，如图 8-191 所示，弹出如图 8-192 所示界面，单击“Assign name”（分配名称），实际上是把模块的名称“ET200M”下载到 IP 地址为 192.168.0.2 的远程站里。

图 8-191　分配设备名称（1）

在菜单栏中，单击“PLC”→“Ethernet”→“验证设备名称”，如图 8-193 所示，弹出如图 8-194 所示界面，可以看到“Status”（状态）下有一个“√”，表示验证成功。

**【关键点】** 分配设备名称和验证设备名称必须在 STEP 7 与 CPU 处于连接状态下才能进行。

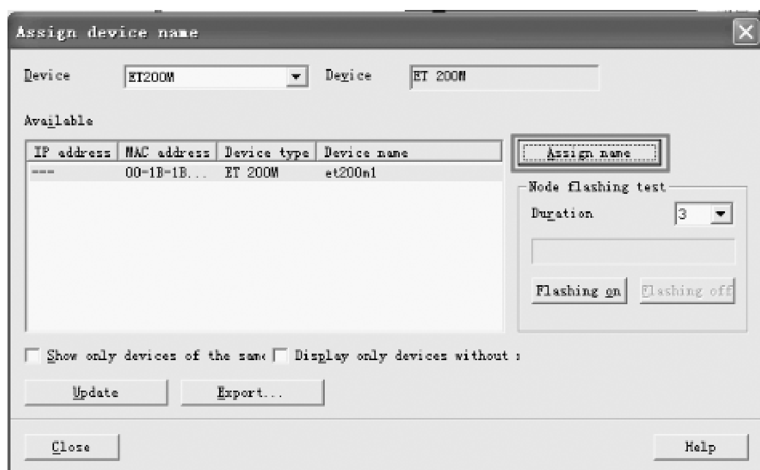


图 8-192　分配设备名称（2）



图 8-193　验证设备名称（1）

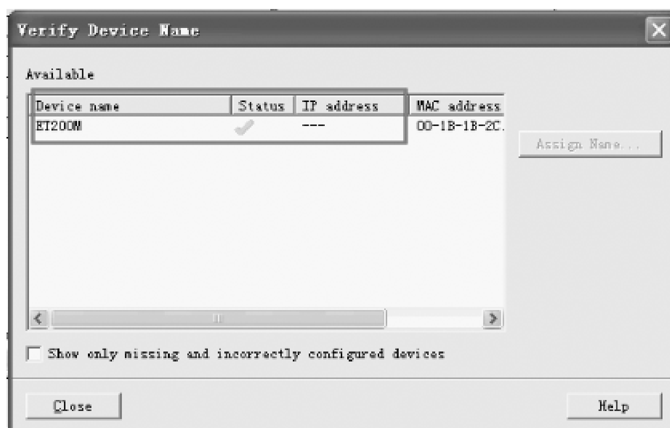


图 8-194　验证设备名称（2）

（3）编写程序

打开 OB1，在 OB1 中编写如图 8-195 所示的梯形图程序。

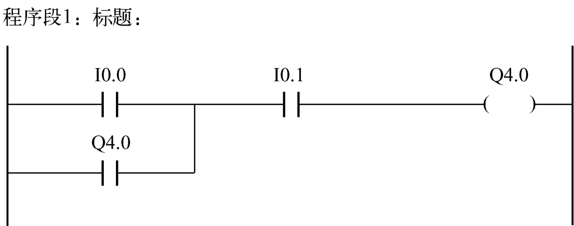



图 8-195　梯形图

### 8.5.4　S7－400 与 S7－200 SMART 间的以太网通信

S7－200 SMART 是西门子公司推出的小型 PLC 即将取代 S7－200。S7－200 SMART 内置 PROFINET 接口（PN 口），S7－400 与 S7－200 SMART 间的以太网通信，可以利用 S7－200 SMART 内置 PN 口，采用 S7 通信协议。以下通过例子介绍 S7－400 与 S7－200 SMART 间的以太网通信。

**【例 8-13】**某系统的控制器由 S7－400、SM421、CP443－1 和 CPU ST40 组成，要将 S7－400 上的 2 个字节 MB0 和 MB1 传送到 CPU ST40 的 MB0 和 MB1，将 CPU ST40 上的 2 个字节 MB10 和 MB11 传送到 S7－400 的 MB10 和 MB11，请组态并编写相关程序。

**解：**以 S7－400 作客服端，S7－200 SMART 作服务器端。

（1） 主要软硬件配置

① 1 套 STEP 7 V5.5 SP4。

② 1 台 CP421－1。

③ 1 台 SM421 和 SM422。

④ 1 台 CP443－1 和 PS407。

⑤ 1 台 CPU ST40。

⑥ 两根网线。

PROFINET 现场总线硬件配置如图 8-196 所示。

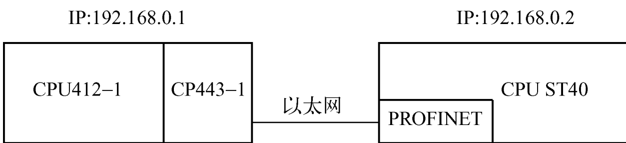



图 8-196　PROFINET 现场总线硬件配置

（2） 硬件组态过程

1） 新建项目和组态硬件。新建项目命名为“PN_SMART”，插入站点 CPU 400，双击

“硬件”，打开硬件组态界面，先插入机架 UR2，再插入电源 PS 407 4A，然后插入 CP443 - 1、DO32 和 DI32 模块，如图 8-197 所示。



图 8-197　新建项目和硬件组态

2）设置客户端 IP 地址。双击如图 8-197 所示的“PN - IO”，打开“PN - IO”的属性界面，单击“属性”按钮，如图 8-198 所示，弹出如图 8-199 所示的界面，设置如图所示的 IP 地址，单击“确定”按钮即可。



图 8-198　PN - IO 属性

3）网络组态。选中如图 8-200 所示的“1”处，单击鼠标右键，弹出快捷菜单，单击“插入新连接”选项，弹出如图 8-201 所示的对话框，选中“未指定”选项和“S7 连接”，




图 8-199　设置客户端 IP 地址

单击“应用”按钮，弹出如图 8-202 的所示界面。

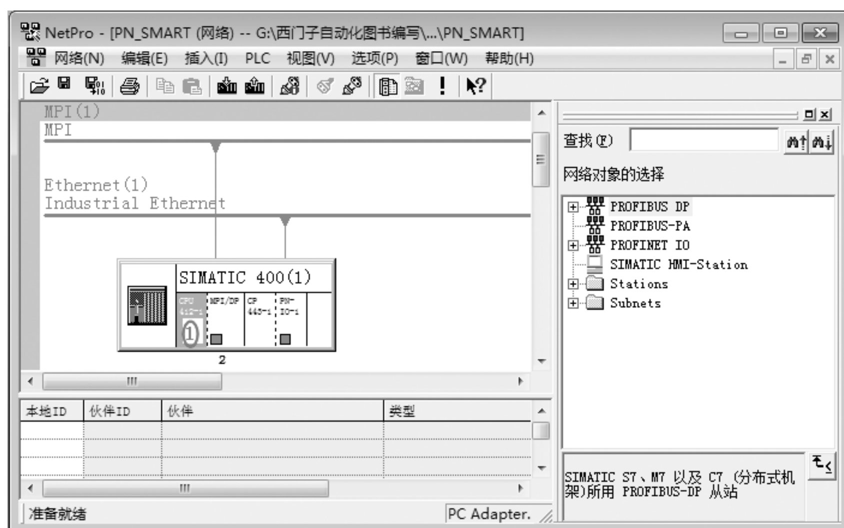



图 8-200　插入新连接（1）

由于 S7－400 是客户端，也就是主控端，本地连接端点勾选“建立主动连接”，如图 8-202 所示；设置伙伴，即服务器端的 IP 地址是“192.168.0.2”；注意，本地 ID 为“1”，这是连接号，在编写程序时要用到；最后单击“地址详细信息”按钮，弹出如图 8-203 所示的界面。

如图 8－203 所示，设置伙伴（S7－200 SMART）的 TSAP 为“03.01”，S7－400 的 TSAP 不变，这一步容易忽略，单击“确定”按钮。最后单击网络组态界面的工具栏的“保存和编译”按钮。




图 8-201　插入新连接（2）




图 8-202　属性 - S7 连接

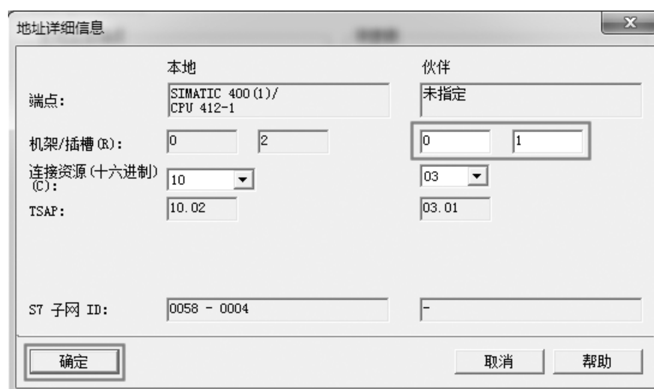



图 8-203　设置详细地址信息

（3）编写梯形图程序

由于 S7－400 作客服端，S7－200 SMART 作服务器端，所以 S7－200 SMART 不需要编写通信程序，只需在 S7－400 中编写程序，如图 8-204 所示。注意 M20.5 为秒脉冲，在硬件组态中设置。




图 8-204　客户端梯形图

## 8.6　网络路由功能

### 8.6.1　网络路由功能概述

在一个自动化系统中，并不是所有的站点都在一个具有相同通信协议的网络中，很多系统同时使用 MPI、PROFIBUS – DP 和工业以太网等多种通信协议网络。一般情况下，是不能跨网络访问站点。在这种情况下，如果 SIMATIC 站具有合适的多个接口可以连接到不同的子网，则可以作为路由设备实现跨网的通信。

要实现不同网络间路由器功能，模块必须具备路由功能。西门子具有路由功能的模块主要包含两类：CPU 模块和 CP 通信模块。多数 CPU 模块具有路由功能（如 CPU314C – 2DP 和 CPU412 – 1），但早期的 CPU 模块没有路由功能。CP 模块（如 CP5611、CP342 – 5、CP343 – 1 和 CP443 – 1 等）通常都具有路由功能。

查看一个模块是否具有路由器功能很简单，在 STEP 7 软件的 SIMATIC Manager 或者硬件组态界面中，选中这个模块，单击鼠标右键，弹出快捷菜单，单击“对象属性”，弹出如图 8–205 对话框，选中“常规”选项卡，可以看到“简单描述：”中有“routing”，这说明这个模块有路由器功能。

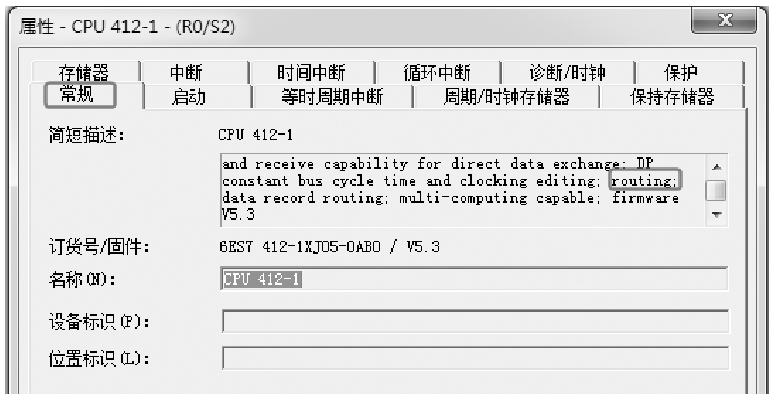



图 8–205　CPU 属性

### 8.6.2　网络路由功能应用

在复杂网络中，不同的网络连接不同的设备。而设备调试和后期的设备维护过程中，我们希望在网络的任意一点（如中控室），对所有的网络站点进行编程调试。早期的产品不支持不同的网络间的通信，现在通过网络路由可以实现这个功能了。下面用一个例子介绍网络路由功能应用。

**【例 8–14】**有一个控制系统由 CPU412 – 1、CP443 – 1、CPU314C – 2P、CP343 – 1 Lean 和 MP270 组成，CPU412 – 1 和 CPU314C – 2P 采用以太网通信，CPU314C – 2P 和 MP270 采用 PROFIBUS – DP 通信，要求在中控室用一台笔记本电脑（配 STEP 7 和 WinCC flexible 软件），通过 PC Adapter（USB 接口的适配器）下载程序，并调试和监控这个系统。

**解：**

（1）新建项目并进行硬件组态

新建项目，命名为“Routing”，先后组态 S7 －400、S7 －300 和 HMI，建立 S7 －400、S7 －300 以太网连接；在组态 S7 －400 时，需新建 MPI 网络，S7 －400 的 MPI 站地址为“2”；在组态 S7 －300，需新建 PROFIBUS 网络，S7 －300 的 DP 站地址为“2”。组态后界面如图 8-206 所示。

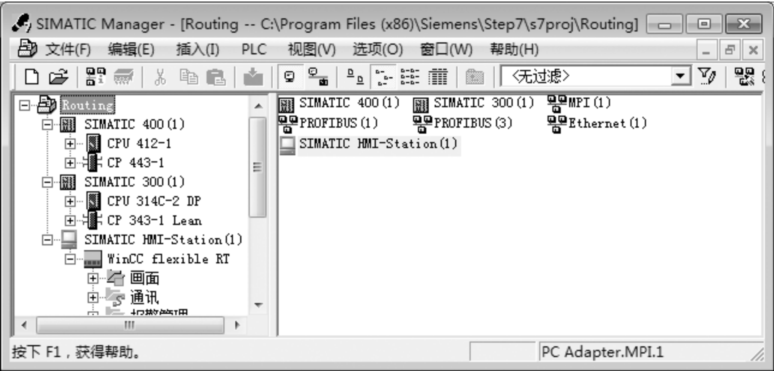



图 8-206　新建项目

（2）网络组态

在 SIMATIC Manager 界面中，单击工具栏的“组态网路”按钮，打开组态网络界面，如图 8-207 所示。此时，把 PC Adapter 连接在 CPU412 －1 上，仅能与 CPU412 －1 通信，不能与 MP270 和 CPU314 －2DP 通信。

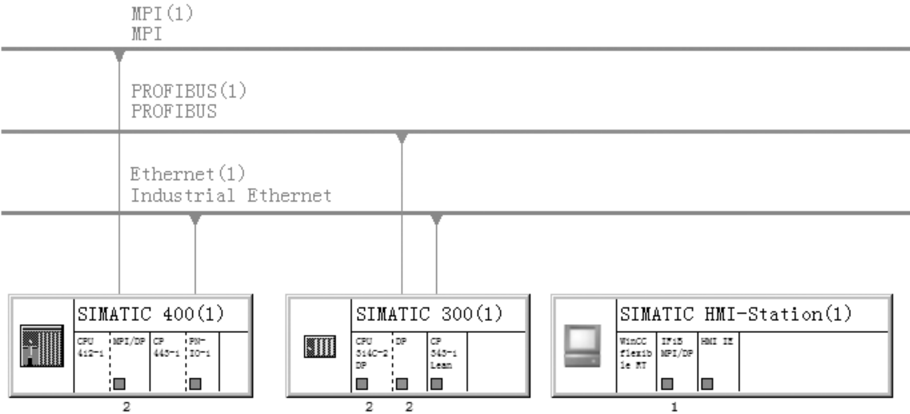



图 8-207　网络组态（1）

在网络组态界面中，双击图 8-207 中 SIMATIC HMI －Station(1)中的红色方框，弹出如图 8-208 所示的选项卡，选择通信接口的类型为“PROFIBUS”；单击“属性”按钮，选中“PROFIBUS(1)　1.5 Mbps”网络，把 HMI 的 DP 地址修改为“1”，单击“确定”按钮，再单击“确定”按钮，此时在网络组态界面可以看到，HMI 已经连接到 PRIFIBUS 网络上了。到目前为止，仍然不能实现路由功能。




图 8-208　网络组态（2）

在网络组态界面中，双击硬件目录中的“PG/PC”，“PG/PC”站点弹出在如图 8-209 中。双击“PG/PC”，弹出“属性 - PG/PC”对话框，如图 8-210 所示，选中“接口”选项，单击“新建”按钮，弹出“新建接口 - 接口类型”对话框，选中“MPI”，单击“确定”按钮。此时，弹出“属性 - MPI 接口”对话框，选择 MPI 地址为“1”，单击“确定”按钮，如图 8-211 所示。

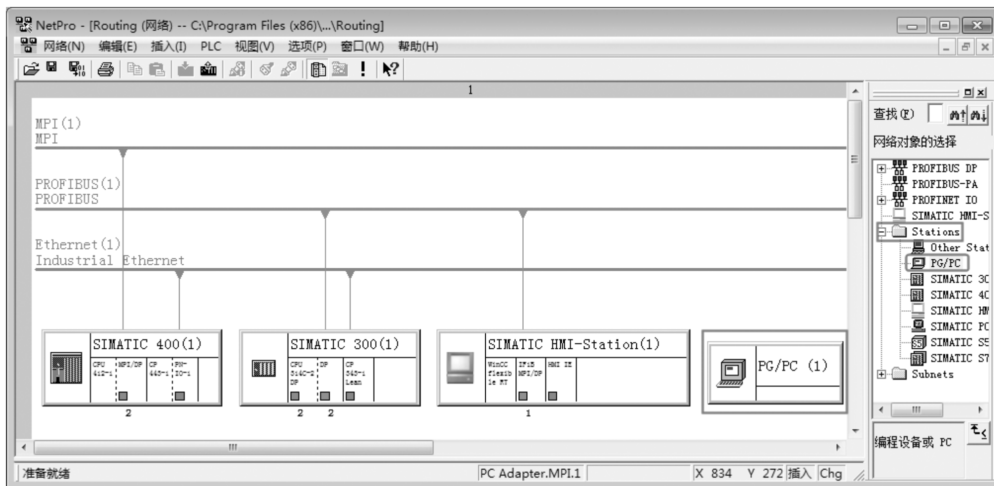



图 8-209　网络组态（3）

如图 8-212 所示，选择“分配”选项卡，选定“PC Adapter. MPI. 1”，单击“确定”按钮，弹出如图 8-213 所示，可以看到“S7ONLINE”下面的激活上已经“勾选”。单击“确定”按钮。在网络组态界面的工具栏，单击“保存和编译”按钮，此时，网络组态如图 8-214 所示，网络组态完成，至此，系统具备路由功能，可以用于下载程序和监控 PLC 了。例如要从 PG/PC 下载到 HMI，其通信路径为：“PG/PC”→“SIMATIC 400(1)”→“SIMATIC 300(1)”→“SIMATIC HMI Station(1)”。




图 8-210　网络组态（4）

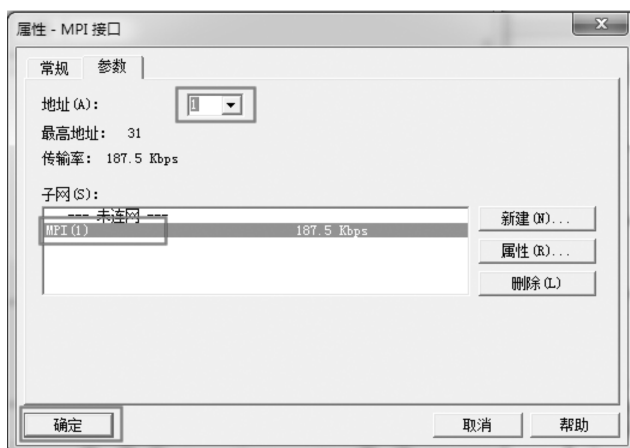



图 8-211　网络组态（5）

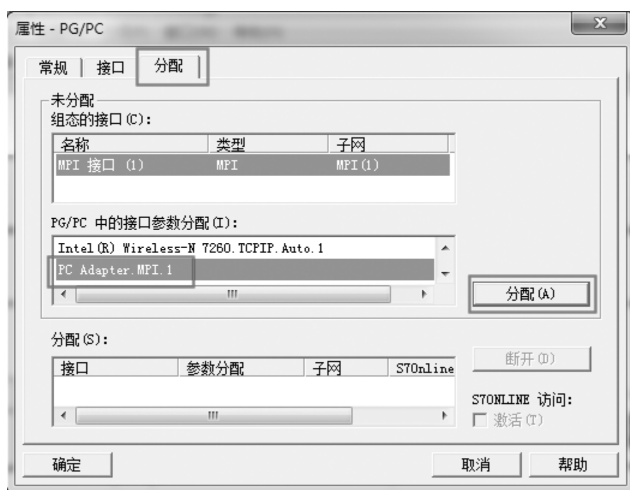



图 8-212　网络组态（6）

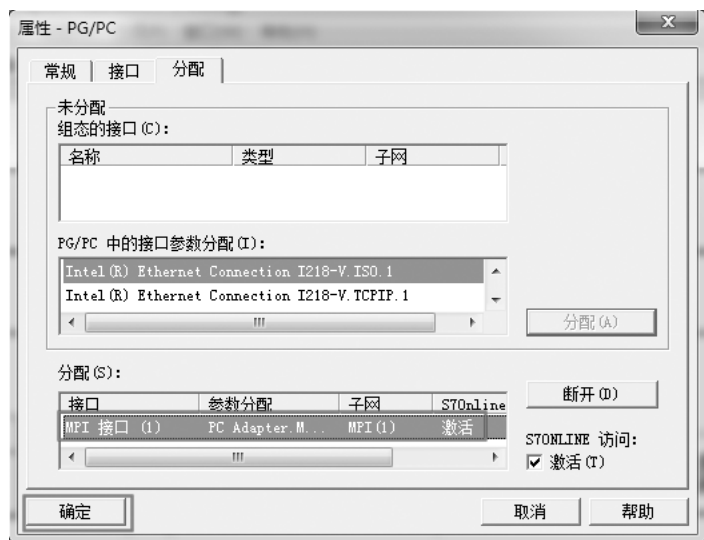



图 8-213　网络组态（7）

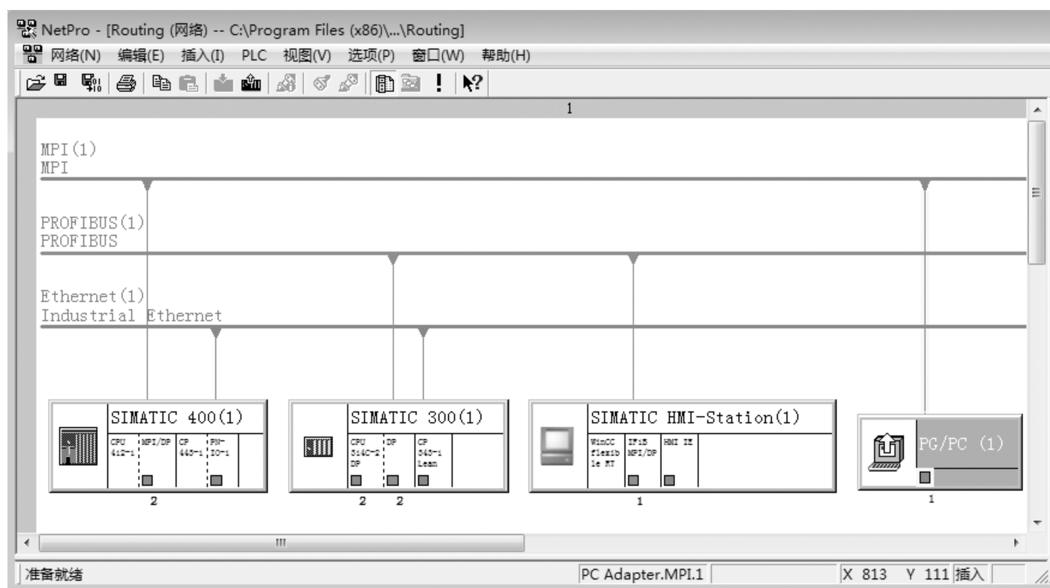



图 8-214　网络组态（8）

（3）下载项目到 HMI

在 SIMATIC Manager 界面中，如图 8-215 所示，选中“连接”，并双击“连接”，打开 WinCC flexible 软件，并在此软件中创建项目，在此不再赘述。

创建完成项目后，单击“选择设备进行传输”按钮，弹出“选择设备进行传输”，如图 8-216 所示，模式选择为“MPI/DP”，勾选“启用路由器”选项，单击“传输”按钮，即可向 HMI 传输项目了。

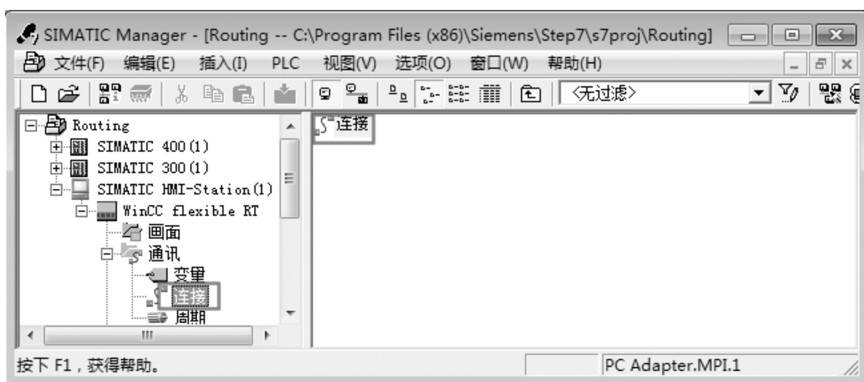



图 8-215　打开 HMI 项目




图 8-216　选择设备进行传输

# 第 9 章　S7－300/400 PLC 的 PID 控制技术

本章介绍 PID 控制的基本原理以及 PID 控制在电炉温度控制中的应用。

## 9.1　PID 控制简介

### 9.1.1　PID 控制原理简介

在过程控制中，按偏差的比例（P）、积分（I）和微分（D）进行控制的 PID 控制器（也称 PID 调节器）是应用最广泛的一种自动控制器。它具有原理简单、易于实现、适用面广、控制参数相互独立、参数选定比较简单以及调整方便等优点；而且在理论上可以证明，对于过程控制的典型对象——“一阶滞后＋纯滞后”与“二阶滞后＋纯滞后”的控制对象，PID 控制器是一种最优控制。PID 调节规律是连续系统动态品质校正的一种有效方法，它的参数整定方式简便，结构改变灵活（如可为 PI 调节、PD 调节等）。长期以来，PID 控制器被广大科技人员及现场操作人员所采用，并积累了大量的经验。

PID 控制器就是根据系统的误差，利用比例、积分和微分计算出控制量来进行控制。当被控对象的结构和参数不能完全掌握或得不到精确的数学模型时，控制理论的其他技术难以采用时，系统控制器的结构和参数必须依靠经验和现场调试来确定，这时应用 PID 控制技术最为方便。即当不完全了解一个系统和被控对象或不能通过有效的测量手段来获得系统参数时，最适合采用 PID 控制技术。

**1. 比例（P）控制**

比例控制是一种最简单、最常用的控制方式，如放大器、减速器和弹簧等。比例控制器能立即成比例地响应输入的变化量。但仅有比例控制时，系统输出存在稳态误差（Steady－state Error）。

**2. 积分（I）控制**

在积分控制中，控制器的输出量是输入量对时间的积累。对于一个自动控制系统，如果在进入稳态后存在稳态误差，则称这个控制系统是有稳态误差的或简称有差系统（System with Steady－state Error）。为了消除稳态误差，在控制器中必须引入“积分项”。积分项对误差的运算取决于时间的积分，随着时间的增加，积分项会增大。所以即便误差很小，积分项也会随着时间的增加而加大，它推动控制器的输出增大，使稳态误差进一步减小，直到等于零。因此，采用比例＋积分（PI）控制器，可以使系统在进入稳态后无稳态误差。

**3. 微分（D）控制**

在微分控制中，控制器的输出与输入误差信号的微分（即误差的变化率）成正比关系。自动控制系统在克服误差的调节过程中可能会出现振荡甚至失稳。其原因是存在有较大的惯性组件（环节）或有滞后组件，具有抑制误差的作用，其变化总是落后于误差的变化。解

决的办法是使抑制误差的作用的变化“超前”，即在误差接近零时，抑制误差的作用就应该是零。这就是说，在控制器中仅引入“比例”项往往是不够的，比例项的作用仅是放大误差的幅值，因而需要增加的是“微分项”，它能预测误差变化的趋势，这样具有比例＋微分的控制器就能够提前使抑制误差的控制作用等于零，甚至为负值，从而避免被控量的严重超调。所以对有较大惯性或滞后的被控对象，比例＋微分（PD）控制器能改善系统在调节过程中的动态特性。

**4. 闭环控制系统特点**

控制系统一般包括开环控制系统和闭环控制系统。开环控制系统（Open－loop Control System）是指被控对象的输出（被控制量）对控制器（Controller）的输出没有影响。在这种控制系统中，不依赖将被控制量反送回来以形成任何闭环回路。闭环控制系统（Closed－loop Control System）的特点是：系统被控对象的输出（被控制量）会反送回来影响控制器的输出，形成一个或多个闭环。闭环控制系统有正反馈和负反馈，若反馈信号与系统给定值信号相反，则称为负反馈（Negative Feedback）；若极性相同，则称为正反馈。一般闭环控制系统均采用负反馈，又称负反馈控制系统。可见，闭环控制系统性能远优于开环控制系统。

**5. PID 控制器的主要优点**

PID 控制器成为应用最广泛的控制器，它具有以下优点：

1）PID 算法蕴涵了动态控制过程中过去、现在和将来的主要信息，而且其配置几乎最优。其中，比例（P）代表了当前的信息，起纠正偏差的作用，使过程反应迅速。微分（D）在信号变化时有超前控制作用，代表将来的信息。在过程开始时强迫过程进行，过程结束时减小超调、克服振荡，提高系统的稳定性，加快系统的过渡过程。积分（I）代表了过去积累的信息，它能消除静差，改善系统的静态特性。此 3 种作用配合得当，可使动态过程快速、平稳并准确，得到良好的效果。

2）PID 控制适应性好，有较强的鲁棒性，对各种工业场合，都可在不同的程度上应用。特别适于“一阶惯性环节＋纯滞后”和“二阶惯性环节＋纯滞后”的过程控制对象。

3）PID 算法简单明了，各个控制参数相对较为独立，参数的选定较为简单，形成了完整的设计和参数调整方法，很容易为工程技术人员所掌握。

4）PID 控制根据不同的要求，针对自身的缺陷进行了不少改进，形成了一系列改进的 PID 算法。例如，为了克服微分带来的高频干扰的滤波 PID 控制；为克服大偏差时出现饱和超调的 PID 积分分离控制；为补偿控制对象非线性因素的可变增益 PID 控制等。这些改进算法在一些应用场合取得了很好的效果。随着智能控制理论的发展，又形成了许多智能 PID 控制方法。

### 9.1.2 PID 控制的算法和图解

**1. PID 的算法**

PID 控制系统原理框图如图 9-1 所示。

PID 控制器调节输出，保证偏差（e）为零，使系统达到稳定状态，偏差是给定值（SP）和过程变量（PV）的差。PID 控制的原理基于以下公式：

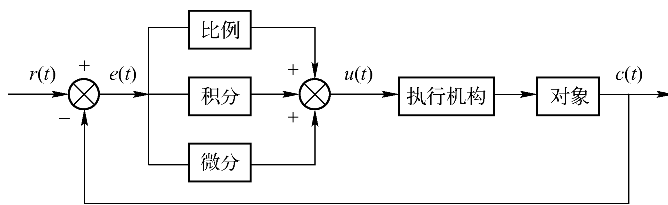



图 9-1　PID 控制系统原理框图

$$M(t)=K_C \cdot e + K_C\int_0^1 e\mathrm{d}t + M_{\mathrm{initial}} + K_C \cdot \frac{\mathrm{d}e}{\mathrm{d}t} \tag{9-1}$$

式（9-1）中，$M(t)$是 PID 回路的输出；$K_C$ 是 PID 回路的增益；$e$ 是 PID 回路的偏差（给定值与过程变量的差）；$M_{\mathrm{initial}}$是 PID 回路输出的初始值。

由于以上的算式是连续量，必须将连续量离散化才能在计算机中运算，离散处理后的算式如下：

$$M_n=K_C \cdot e_n + K_I \cdot \sum_1^n e_x + M_{\mathrm{initial}} + K_D \cdot (e_n - e_{n-1}) \tag{9-2}$$

式（9-2）中，$M_n$ 是在第 n 采样时刻的 PID 回路的输出计算值；$K_C$ 是 PID 回路的增益；$K_I$ 是积分项的比例常数；$K_D$ 是微分项的比例常数；$e_n$ 是采样时刻 $n$ 的回路的偏差值；$e_{n-1}$是采样时刻 $n-1$ 的回路的偏差值；$e_x$ 是采样时刻 $x$ 的回路的偏差值；$M_{\mathrm{initial}}$是 PID 回路输出的初始值。

再对以上算式进行改进和简化，得出如下计算 PID 输出的算式：

$$M_n=MP_n+MI_n+MD_n \tag{9-3}$$

式（9-3）中，$M_n$ 是第 $n$ 采样时刻的计算值；$MP_n$ 是第 $n$ 采样时刻的比例项值；$MI_n$ 是第 $n$ 采样时刻的积分项的值；$MD_N$ 是第 $n$ 采样时刻微分项的值。

$$MP_n=K_C*(SP_n-PV_n) \tag{9-4}$$

式（9-4）中，$MP_n$ 是第 $n$ 采样时刻的比例项值；$K_C$ 是增益；$SP_n$ 是第 $n$ 次采样时刻的给定值；$PV_n$ 是第 $n$ 次采样时刻的过程变量值。很明显，比例项 $MP_n$ 数值的大小和增益 $K_C$ 成正比，增益 $K_C$ 增加可以直接导致比例项 $MP_n$ 的快速增加，从而直接导致 $M_n$ 增加。

$$MI_n=K_C*T_S/T_I*(SP_n-PV_n)+MX \tag{9-5}$$

式（9-5）中，$K_C$ 是增益；$T_S$ 是回路的采样时间；$T_I$ 是积分时间；$SP_n$ 是第 $n$ 次采样时刻的给定值；$PV_n$ 是第 $n$ 次采样时刻的过程变量值；$MX$ 是第 $n-1$ 时刻的积分项（也称为积分前项）。很明显，积分项 $MI_n$ 数值的大小随着积分时间 $T_I$ 的减小而增加，$T_I$ 的减小可以直接导致积分项 $MI_n$ 数值的增加，从而直接导致 $M_n$ 增加。

$$MD_n=K_C*(PV_{n-1}-PV_n)*T_D/T_S \tag{9-6}$$

式（9-6）中，$K_C$ 是增益；$T_S$ 是回路的采样时间；$T_D$ 是微分时间；$PV_n$ 是第 $n$ 次采样时刻的过程变量值；$PV_{n-1}$是第 $n-1$ 次采样时刻的过程变量。很明显，微分项 $MD_n$ 数值的大小随着微分时间 $T_D$ 的增加而增加，$T_D$ 的增加可以直接导致积分项 $MD_n$ 数值的增加，从而直

接导致 $M_n$ 增加。

**【关键点】** 式（9-3）～式（9-6）是非常重要的。根据这几个公式，读者必须建立一个概念：增益 $K_C$ 增加可以直接导致比例项 $MP_n$ 的快速增加，$T_I$ 的减小可以直接导致积分项 $MI_n$ 数值的增加，微分项 $MD_n$ 数值的大小随着微分时间 $T_D$ 的增加而增加，从而直接导致 $M_n$ 增加。理解了这一点，对于正确调节 P、I、D 这三个参数是至关重要的。

**2. PID 的算法图解**

以上 PID 控制进行了数学计算，但对于数学功底薄弱的初学者是较难理解的，以下将用图解对 PID 的参数的作用进行说明，这样更加直观易懂。

（1）纯比例控制曲线

纯比例控制曲线如图 9-2 所示。

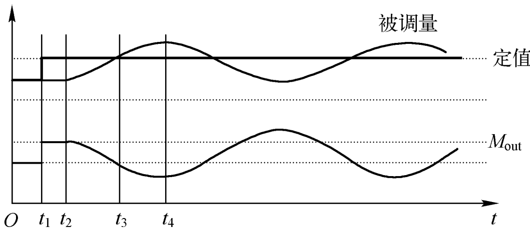



图 9-2　纯比例控制曲线

图 9-2 中，假设被调量（如果是电炉，那么被调量是温度）偏高时，调门应关小，即 PID 为负作用。在设定值有一阶跃扰动时，调节器输入偏差为 $-\Delta e$。此时 $M_{out}$（就是 PID 运算后的输出值）也应有一阶跃量 $\Delta e \cdot K_c$，然后被调量不变。经过一个滞后期 $t_2$，被调量开始响应 $M_{out}$。因为被调量增加，$M_{out}$ 也开始降低。一直到 $t_4$ 时刻，被调量开始回复时，$M_{out}$ 才开始升高。两曲线虽然波动相反，但是图形如果反转，就可以看出是相似形。

（2）纯积分控制曲线

纯积分控制曲线如图 9-3 所示。

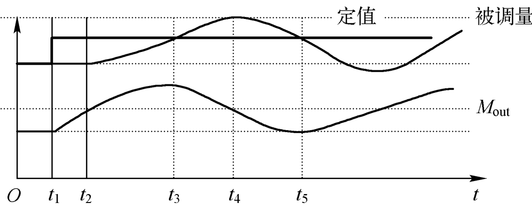



图 9-3　纯积分控制曲线

积分作用下的调节曲线，因输出的响应较比例作用不明显，故被调量开始变化的时刻 $t_2$，较比例作用缓慢。在 $t_1$ 到 $t_2$ 的时间内，因为被调量不变，即输入偏差不变，所以输出以不变的速率上升，即呈线性上升。调节器的输出缓慢改变，导致被调量逐渐受到影响而改变。在 $t_2$ 时刻，被调量开始变化时，输入偏差逐渐减小，输出的速率开始降低。到 $t_3$ 时刻，偏差为 0 时，输出不变，输出曲线为水平。然后偏差开始为正时，输出才开始降低。到 $t_4$ 时刻，被调量达到顶点开始回复，但是因偏差仍旧为正，故输出继续降低只是速率开始减缓。直到 $t_5$ 时刻，偏差为 0 时，输出才重新升高。

(3) 比例积分控制曲线

如图 9-4 所示，设定值有阶跃扰动时，比例作用使输出曲线 $M_{out}$同时有一个阶跃扰动，同时积分作用使 $M_{out}$开始继续增大。$t_2$ 时刻后，被调量响应 $M_{out}$开始增大。此时比例作用因 $\Delta e$ 减小而使 $M_{out}$开始降低（如图中点划线 $M_{out}(\delta)$所示）；但是积分作用与 $\Delta e$ 的趋势无关，与 $\Delta e$ 的正负有关，积分作用因 $\Delta e$ 还在负向，故继续使 $M_{out}$增大，只是速率有所减缓。比例作用和积分作用的叠加，决定了 $M_{out}$的实际走向，如图 $M_{out}(\delta_i)$所示。只要比例作用不是无穷大，或是积分作用不为零，从 $t_2$时刻开始，总要有一段时间是积分作用强于比例作用，使得 $M_{out}$继续升高。然后持平（$t_3$时刻），然后降低。在被调量升到顶峰的 $t_5$时刻，同理，比例作用使 $M_{out}$也达到顶点（负向），而积分作用使得最终 $M_{out}$的顶点向后延时（$t_6$时刻）。

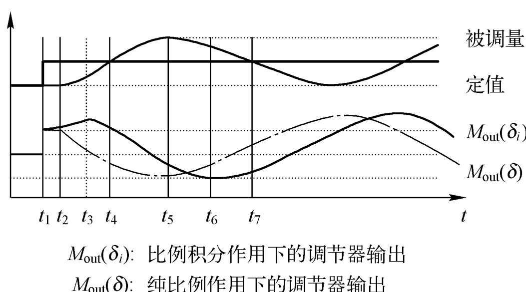



图 9-4　比例积分控制曲线

### 9.1.3　PID 控制器的参数整定

PID 控制器的参数整定是控制系统设计的核心内容。根据被控过程的特性，确定 PID 控制器的比例系数、积分时间和微分时间的大小。PID 控制器参数整定的方法很多，概括起来有如下两大类：

一是理论计算整定法。它主要依据系统的数学模型，经过理论计算确定控制器参数。这种方法所得到的计算数据未必可以直接使用，还必须通过工程实际进行调整和修改。

二是工程整定法。它主要依赖于工程经验，直接在控制系统的试验中进行，且方法简单、易于掌握，在工程实际中被广泛采用。

**1. 整定的方法和步骤**

现在一般采用的是临界比例法。利用该方法进行 PID 控制器参数的整定步骤如下：

1) 首先预选择一个足够短的采样周期使系统工作。

2) 仅加入比例控制环节，直到系统对输入的阶跃响应出现临界振荡，记下这时的比例放大系数和临界振荡周期。

3) 在一定的控制度下通过公式计算得到 PID 控制器的参数。

**2. PID 参数的经验值**

在实际调试中，只能先大致设定一个经验值，然后根据调节效果修改。常见系统的经验值如下：

1）对于温度系统：P（%）20～60，I（分）3～10，D（分）0.5～3 。

2）对于流量系统：P（%）40～100，I（分）0.1～1。

3）对于压力系统：P（%）30～70，I（分）0.4～3。

4）对于液位系统：P（%）20～80，I（分）1～5。

**3. PID 参数的整定实例**

PID 参数的整定对于初学者来说并不容易，不少初学者看到 PID 的曲线往往不知道是什么含义，当然也就不知道如何着手调节了，以下用几个简单的例子进行介绍。

**【例 9-1】**某系统的电炉在进行 PID 参数整定时，其输出曲线如图 9-5 所示，设定值和测量值重合（55℃），所以有人认为 PID 参数整定成功，请读者分析，并给出自己的见解。

**解：**在 PID 参数整定时，分析曲线图是必不可少的，测量值和设定值基本重合这是基本要求，并非说明 PID 的参数整定就一定合理。

分析 PID 运算结果的曲线至关重要。如图 9-5 所示，PID 运算结果的曲线虽然很平滑，但过于平坦，这样电炉在运行过程中，其抗干扰能力弱，也就是说，当负载对热量需要稳定时，这个设定无问题，但当负载热量变化大时，测量值和设定值就未必处于重合状态了。这种 PID 运算结果的曲线过于平坦说明 P 过小。

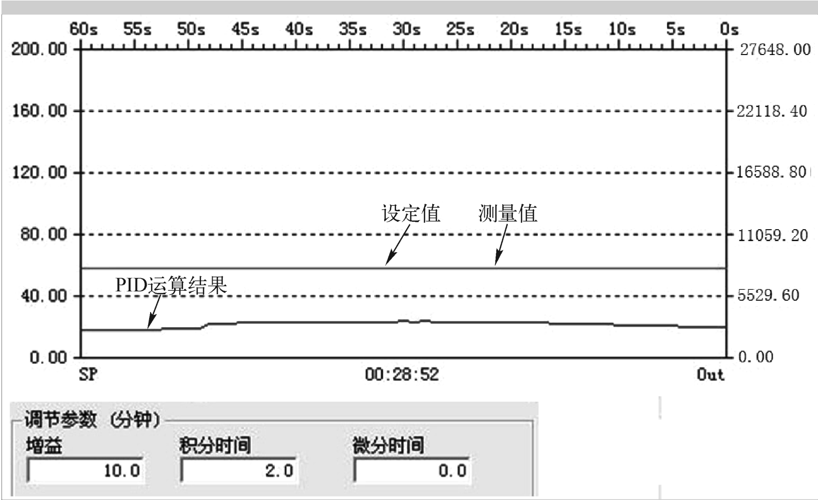



图 9-5　PID 曲线图（1）

将 P 的数值设定为 30.0，如图 9-6 所示，整定就比较合理了。

**【例 9-2】**某系统的电炉在进行 PID 参数整定，其输出曲线如图 9-7 所示，设定值和测量值重合（55℃），所以有人认为 PID 参数整定成功，请读者分析，并给出自己的见解。

**解：**如图 9-7 所示，虽然测量值和设定值基本重合，但 PID 参数整定不合理。

这是因为 PID 运算结果的曲线已经超出了设定的范围，实际就是超调，说明比例环节 P 过大。

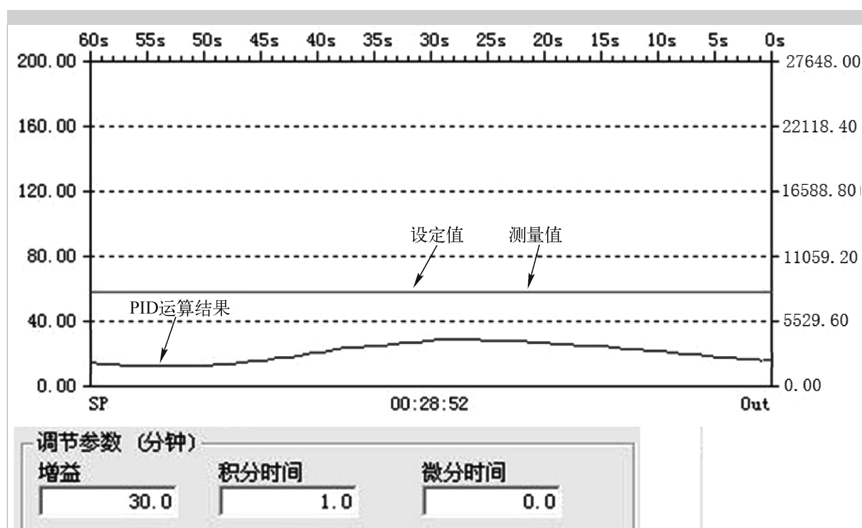



图 9-6　PID 曲线图（2）

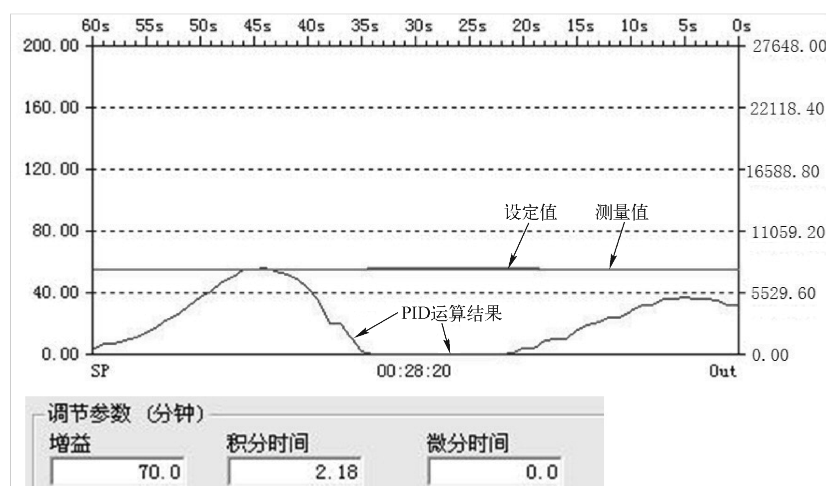



图 9-7　PID 曲线图（3）

## 9.2　利用 S7 -300 进行电炉的温度控制

S7 -300 PLC 提供有 PID 控制功能块来实现 PID 控制。STEP 7 提供了系统功能块 FB41/SFB41、FB43/SFB42 及 FB43/SFB43 实现 PID 闭环控制，其中 FB41/SFB41 “CONT_C” 用于连续控制，FB42/SFB42 “CONT_S” 用于步进控制，FB43/SFB43 “PULSEGEN” 用于脉冲宽度调制，它们位于文件夹 “Libraries→Standard Library→PID Controller” 中。位于文件夹 “Libraries→Standard Library→PID Controller” 的 FB41、FB41、FB43 与 SFB41、SFB42、SFB43 兼容，FB58、FB59 则用于 PID 温度控制。它们是系统固化的纯软件控制器，运行过程中循环扫描、计算所需的全部数据存储在分配给 FB 或 SFB 的背景数据块里，因此可以无

限次调用。FB41 用于 S7 -300，而 SFB41 用于 S7 -400 中。

**【例 9-3】**有一台电炉，要求炉温控制在一定的范围。电炉的工作原理如下：

当设定电炉温度后，CPU 314C -2DP 经过 PID 运算后由自带模拟量输出模块输出一个电压信号送到控制板，控制板根据电压信号（弱电信号）的大小控制电热丝的加热电压（强电）的大小（甚至断开），温度传感器测量电炉的温度，温度信号经过控制板的处理后输入到模拟量输入模块，再送到 CPU 314C -2DP 进行 PID 运算，如此循环。整个系统的硬件配置如图 9-8 所示。请编写控制程序。

**解：**

（1）主要软硬件配置

① 1 套 STEP 7 V5. 5 SP4。

② 1 台 CPU 314C -2DP。

③ 1 根编程电缆。

④ 1 台电炉（含控制板）。

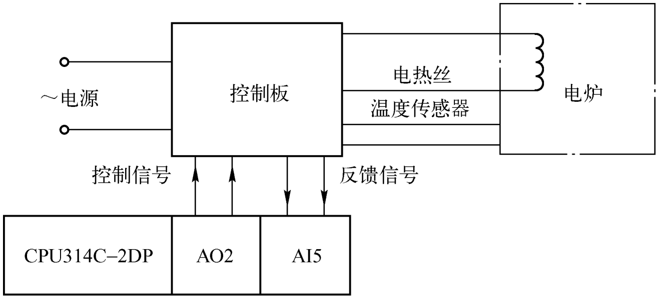



图 9-8　硬件配置图

电气原理图如图 9-9 所示，注意：控制板采集到热电偶的信号，并将其处理成 0 ~ 10 V 的信号送到 CPU314C -2DP 的模拟量输入端子 2 和 4 上。CPU314C -2DP 的模拟量输出端子 16 和 20，将 0 ~ 10 V 的信号送到控制板，控制板将根据其大小，转换成通断信号控制加热

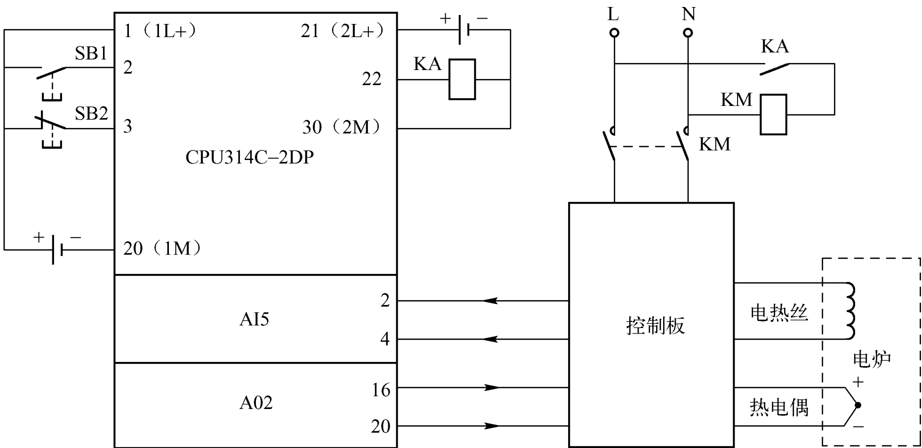



图 9-9　电气原理图

强度。控制板的内部结构不在本书的讨论范围，请读者参看其他资料。

（2）硬件组态

1）新建项目，并插入站点。新建项目命名为“PID”，插入站点“SIMATIC 300（1）”，再在“块”里插入组织块“OB35”“OB100”和参数表“VAT_1”，如图9-10所示。

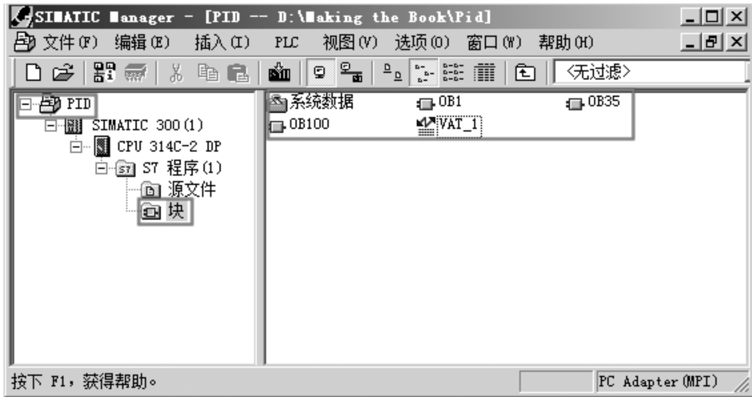



图9-10　新建项目并插入站点

2）更改地址。双击“DI24/DO16”，将数字量输入/输出的起始地址修改成从0开始，双击“AI5/AO2”，将模拟量输入/输出的起始地址修改成从4开始，如图9-11所示。修改地址后，编写梯形图程序时，要与此地址对应。

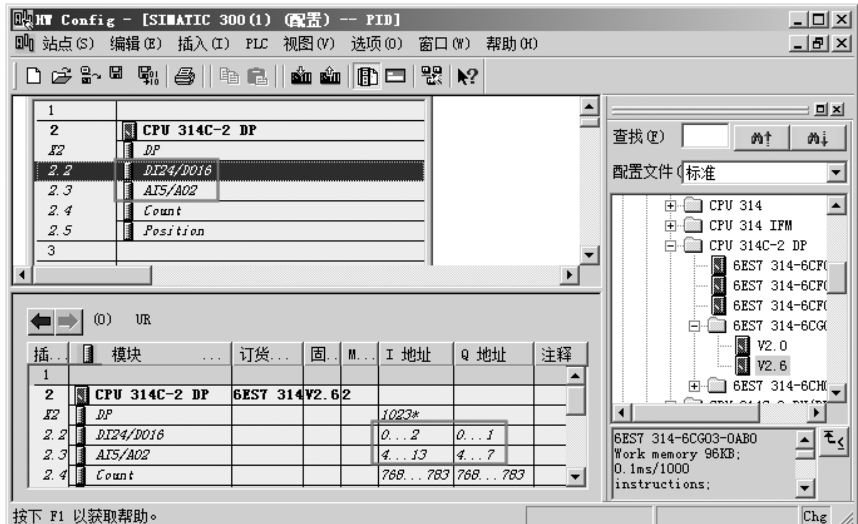



图9-11　更改地址

3）设置模拟输入量测量范围。先选定“输入”选项卡，选择通道0的测量范围为“0～10 V”，其余未使用的通道取消激活，最后单击“确定”按钮，如图9-12所示。

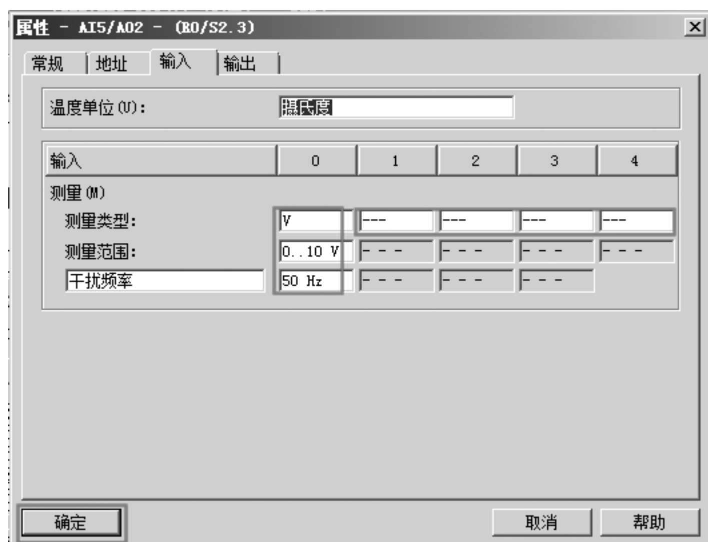



图 9-12　设置模拟输入量测量范围

4）设置模拟输出量测量范围。先选定“输出”选项卡，再选择通道 0 的输出电压范围为“0 ~ 10 V”，其余未使用的通道取消激活，单击“确定”按钮，如图 9-13 所示。

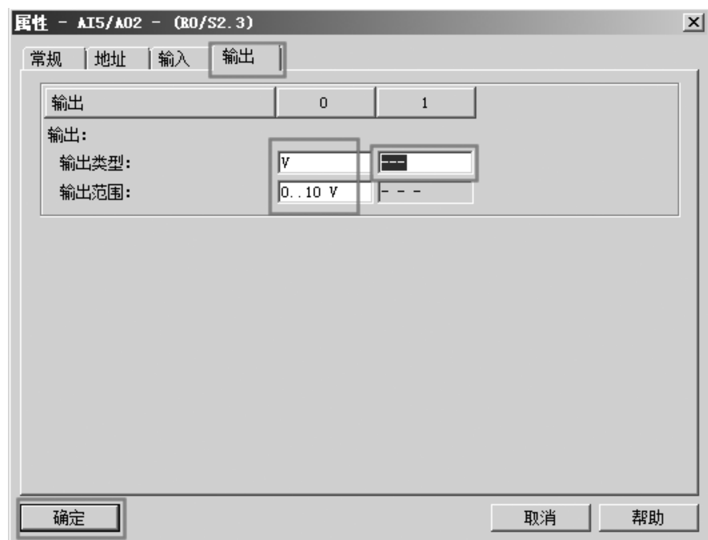



图 9-13　设置模拟输出量测量范围

（3）相关指令介绍

1）FB41 指令。FB41“CONT_C”（连续控制器）在 SIMATIC S7 可编程序逻辑控制器上使用，通过持续的输入和输出变量来控制工艺过程。在参数分配期间，可以通过激活或取消激活 PID 控制器的子功能，使控制器适应工艺过程的需要。

可以使用该控制器作为 PID 固定设定值控制器，或在多循环控制中作为层叠、混料或比

率控制器。该控制器的功能基于使用模拟信号的采样控制器的 PID 控制算法，必要时可以通过加入脉冲发生器阶段进行扩展，为使用成比例执行机构的两个或三个步骤控制器生成脉冲持续时间调制输出信号。但要注意只有在以固定时间间隔调用块时，在控制块中计算的值才是正确的。为此，应该在周期性中断 OB（OB30 ~ OB38）中调用控制块。在 CYCLE 参数中输入采样时间。FB41 指令的主要参数见表 9-1。

**表 9-1　FB41 指令参数**

| LAD | 输入/输出 | 含　　义 | 数 据 类 型 |
| --- | --- | --- | --- |
| FB41<br>EN　ENO<br>COM_RST　LMN<br>MAN_ON　LMN_PER<br>PVPER_ON　QLMN_HLM<br>P_SEL　QLMN_LLM<br>I_SEL　LMN_P<br>INT_HOLD　LMN_I<br>I_ITL_ON　LMN_D<br>D_SEL　PV<br>CYCLE　ER<br>SP_INT<br>PV_IN<br>PV_PER<br>MAN<br>GAIN<br>TI<br>TD<br>TM_LAG<br>DEADB_W<br>LMN_HLM<br>LMN_LLM<br>PV_FAC<br>PV_OFF<br>LMN_FAC<br>LMN_OFF<br>I_ITLVAL<br>DISV | EN | 使能 | BOOL |
| | COM_RST | 为 1 时，重新起动 PID，复位 PID 内部参数 | BOOL |
| | MAN_ON | 为 1 时控制循环中断，直接将 MAN 的值送到 LMN | BOOL |
| | PVPER_PV | 为 1 时，使用 I/O 输入的过程变量 | BOOL |
| | P_SEL | 为 1 时，打开比例 P 操作 | BOOL |
| | I_SEL | 为 1 时，打开积分 I 操作 | BOOL |
| | INT_HOLD | 为 1 时，积分输出被冻结 | BOOL |
| | I_ITL_ON | 为 1 时，I_ITLVAL 作为积分初值 | BOOL |
| | D_SEL | 为 1 时，打开微分 D 操作 | BOOL |
| | CYCLE | 采样时间 | TIME |
| | SP_INT | PID 给定值 | REAL |
| | PV_IN | 浮点格式过程变量输入 | REAL |
| | PV_PER | I/O 格式过程变量输入 | WORD |
| | MAN | 手动值 | REAL |
| | GAIN | 比例增益，用于设置控制器的增益 | REAL |
| | TI | 积分时间输入，积分响应时间 | TIME |
| | TD | 微分时间输入，微分响应时间 | TIME |
| | TM_LAG | 微分操作的延时时间输入 | TIME |
| | DEADB_W | 死区宽度 | REAL |
| | LMN_HLM | 控制器输出上限 | REAL |
| | LMN_LLM | 控制器输出下限 | REAL |
| | PV_FAC | 输入过程变量的比例因子 | REAL |
| | PV_OFF | 输入过程变量的偏移量 | REAL |
| | LMN_FAC | 输出过程变量的比例因子 | REAL |
| | LMN_OFF | 输出过程变量的偏移量 | REAL |
| | I_ITLVAL | 积分操作的初始值 | REAL |
| | DISV | 允许扰动量，一般不设置 | REAL |
| | LMN | 浮点格式的 PID 输出值 | REAL |

（续）

| LAD | 输入/输出 | 含　　义 | 数据类型 |
|---|---|---|---|
| | LMN_PER | I/O 格式的 PID 输出值 | WORD |
| | QLMN_HLM | PID 输出值超出上限 | BOOL |
| | QLMN_LLM | PID 输出值超出下限 | BOOL |
| | LMN_P | PID 输出值中的比例成分 | REAL |
| | LMN_I | PID 输出值中的积分成分 | REAL |
| | LMN_D | PID 输出值中的微分成分 | REAL |
| | PV | 格式化的过程变量输出 | REAL |
| | ER | 死区处理后的误差输出 | REAL |

2）FC105 指令。SCALE 功能接受一个整型值（IN），并将其转换为以工程单位表示的介于下限和上限（LO_LIM 和 HI_LIM）之间的实型值。将结果写入 OUT。SCALE 功能使用以下等式：

$$OUT = (FLOAT(IN) - K1)/(K2 - K1) \times (HI_LIM - LO_LIM)] + LO_LIM$$

常数 K1 和 K2 根据输入值是 BIPOLAR（双极性）还是 UNIPOLAR（单极性）设置。BIPOLAR 的含义是：假定输入整型值介于 −27 648 与 27 648 之间，则 K1 = −27 648.0，K2 = +27 648.0。UNIPOLAR 的含义是：假定输入整型值介于 0 ~ 27 648 之间，则 K1 = 0.0，K2 = +27 648.0。FC105 指令参数见表 9-2。

**表 9-2　FC105 指令参数**

| LAD | 输入/输出 | 含　　义 | 数据类型 |
|---|---|---|---|
| "SCALE"<br>EN　ENO<br>IN　RET_VAL<br>HI_LIM　OUT<br>LO_LIM<br>BIPOLAR | EN | 使能，信号状态为 1 时激活该功能 | BOOL |
| | HI_LIM | 常数，工程单位表示的上限值 | REAL |
| | LO_LIM | 常数，工程单位表示的下限值 | REAL |
| | BIPOLAR | 为 1 时，表示输入值为双极性。信号状态 0 表示输入值为单极性 | BOOL |
| | IN | 输入值 | INT |
| | OUT | 转换的结果 | REAL |
| | RET_VAL | 如果该指令的执行没有错误，将返回值 W#16#0000。对于 W#16#0000 以外的其他值，参见“错误信息” | WORD |

3）FC106 指令。UNSCALE 功能指令将一个从低限 LO_LIM 到高限 HI_LIM 工程单位的数值转换成一个整数值，将结果写入 OUT 中。这个指令满足如下公式：

$$OUT = (IN - LO_LIM)/(HI_LIM - LO_LIM) * (K2 - K1) + K1$$

常数 K1 和 K2 根据输入值是 BIPOLAR（双极性）还是 UNIPOLAR（单极性）设置。BIPOLAR 的含义是假定输出整型值介于 −27 648 与 27 648 之间，则 K1 = −27 648.0，K2 = +27 648.0。UNIPOLAR 的含义是假定输出整型值介于 0 ~ 27 648 之间，则 K1 = 0.0，K2 = +27 648.0。FC106 指令参数见表 9-3。

**表 9-3　FC106 指令参数**

| LAD | 输入/输出 | 含　　义 | 数 据 类 型 |
|---|---|---|---|
| "UNSCALE"<br>EN　ENO<br>IN　RET_VAL<br>HI_LIM　OUT<br>LO_LIM<br>BIPOLAR | EN | 使能，信号状态为 1 时激活该功能 | BOOL |
| | HI_LIM | 常数，工程单位表示的上限值 | REAL |
| | LO_LIM | 常数，工程单位表示的下限值 | REAL |
| | BIPOLAR | 为 1 时，表示输入值为双极性。信号状态 0 表示输入值为单极性 | BOOL |
| | IN | 要转换的输入值 | REAL |
| | OUT | 转换的结果 | INT |
| | RET_VAL | 如果该指令的执行没有错误，将返回值 W#16#0000。对于 W#16#000 以外的其他值，参见“错误信息” | WORD |

（4）编写程序

在符号编辑器中输入各个地址对应的符号，如图 9-14 所示，这是编写大型程序时，不能缺少的步骤。

符号编辑器 - [S7 程序(1) (符号) -- PID\SIMATIC 300(1)\CPU 314C-2 DP]

符号表(S)　编辑(E)　插入(I)　视图(V)　选项(O)　窗口(W)　帮助(H)

全部符号

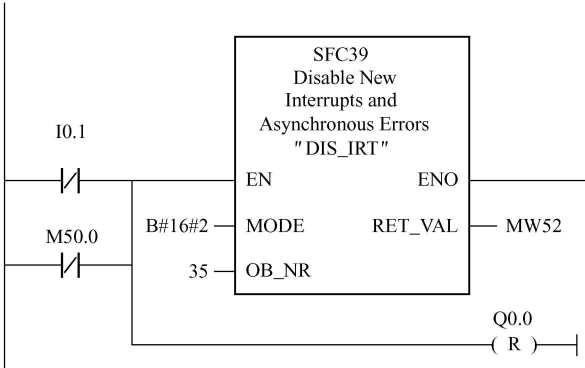
| | 状态 | 符号 | 地址 | | 数据类型 | | 注释 |
|---|---|---|---|---|---|---|---|
| 1 | | COMPLETE RESTART | OB | 100 | OB | 100 | Complete Restart |
| 2 | | CONT_C | FB | 41 | FB | 41 | Continuous Control |
| 3 | | CYC_INT5 | OB | 35 | OB | 35 | Cyclic Interrupt 5 |
| 4 | | DIS_IRT | SFC | 39 | SFC | 39 | Disable New Interrupts and Asynchronous Errors |
| 5 | | EN_IRT | SFC | 40 | SFC | 40 | Enable New Interrupts and Asynchronous Errors |
| 6 | | PID结果 | MD | 28 | REAL | | |
| 7 | | SCALE | FC | 105 | FC | 105 | Scaling Values |
| 8 | | UNSCALE | FC | 106 | FC | 106 | Unscaling Values |
| 9 | | VAT_1 | VAT | 1 | | | |
| 10 | | 比例P | MD | 16 | REAL | | |
| 11 | | 比例环节选择 | M | 0.1 | BOOL | | |
| 12 | | 积分I | MD | 20 | TIME | | |
| 13 | | 积分环节选择 | M | 0.2 | BOOL | | |
| 14 | | 手动输入 | MD | 12 | REAL | | |
| 15 | | 手自转换 | M | 0.0 | BOOL | | |
| 16 | | 微分D | MD | 24 | TIME | | |
| 17 | | 微分环节选择 | M | 0.3 | BOOL | | |
| 18 | | 温度测量值 | MD | 8 | REAL | | |
| 19 | | 温度设置 | MD | 4 | REAL | | |
| 20 | | 重启 | M | 0.5 | BOOL | | |
| 21 | | 启动 | I | 0.0 | BOOL | | |
| 22 | | 停止 | I | 0.1 | BOOL | | |
| 23 | | | | | | | |

按下 F1 获取帮助。　NUM

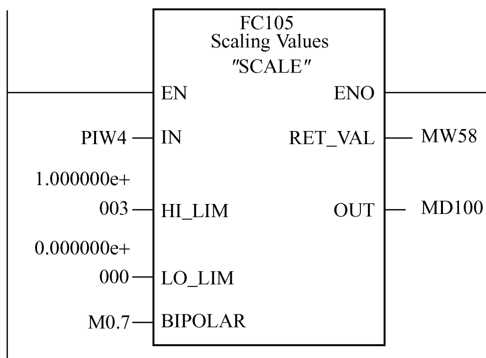
图 9-14　符号编辑器

OB1 中的程序如图 9-15 所示；OB100 中的程序如图 9-16 所示，目的是重启 PID；OB35 中的程序如图 9-17 所示，每 0.1 s 做一次 PID 运算。

程序段1：标题：



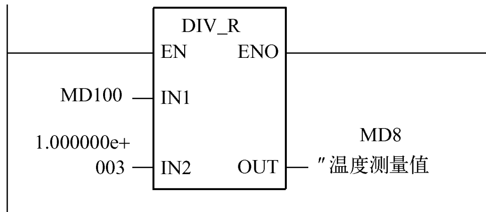

程序段4：模数转换



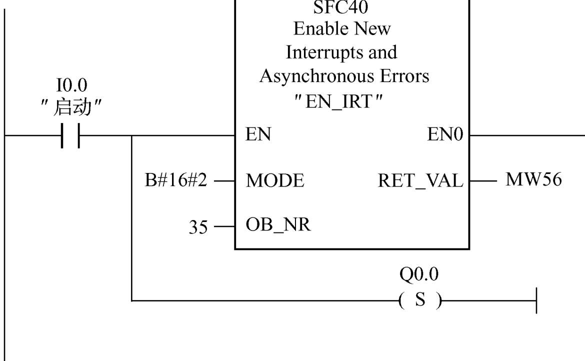

程序段2：标题：




程序段5：规格化




程序段3：标题：




程序段6：标题：

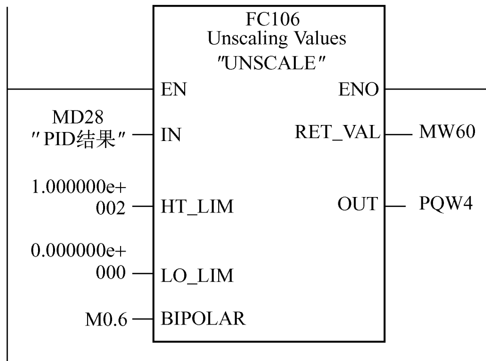



图 9-15　OB1 中的程序

**程序段 1：标题：**

S　"重启"　M0.5

**程序段 2：标题：**

R　"重启"　M0.5

图 9-16　OB100 中的程序

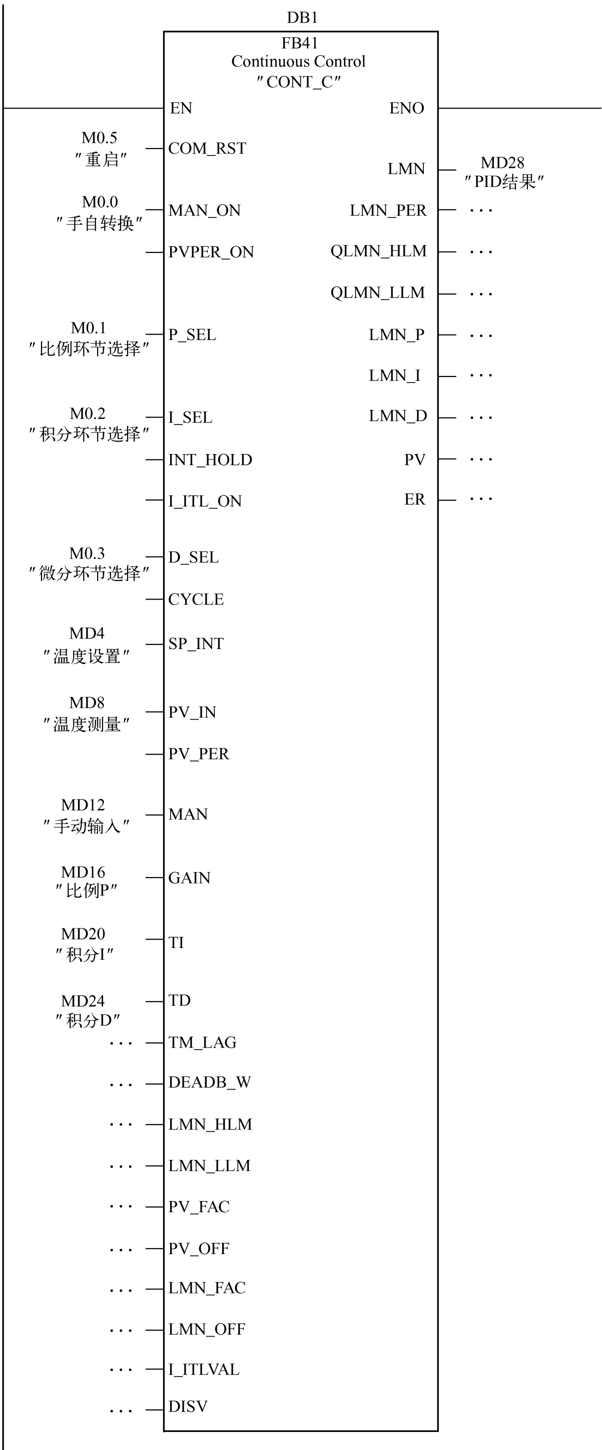



图 9-17　OB35 中的程序

# 第 10 章　西门子人机界面（HMI）应用

本章主要介绍人机界面的入门知识，并介绍如何完成一个简单人机界面项目的过程。

## 10.1　人机界面简介

### 10.1.1　认识人机界面

人机界面（Human Machine Interface，HMI）又称人机接口，在控制领域，HMI 一般特指用于操作员与控制系统之间进行对话和相互作用的专用设备，中文名称为触摸屏。触摸屏技术是起源于 20 纪 70 年代出现的一项新的人机交互作用技术。利用触摸屏技术，用户只需轻轻触碰计算机显示屏上的文字或图符就能实现对主机的操作，部分取代或完全取代键盘和鼠标。它作为一种新的计算机输入设备，是目前最简单、自然和方便的人机交互方式。当下，触摸屏已经在消费电子（如手机）、银行、税务、电力、电信和工业控制等部门得到了广泛的应用。

**1. 触摸屏的工作原理**

触摸屏工作时，用手或其他物体碰触触摸屏，系统会根据手指触摸的图标或文字的位置来定位选择信息输入。触摸屏由触摸检测器件和触摸屏控制器组成。触摸检测部件安装在显示器的屏幕上，用于检测用户触摸的位置，接收后送至触摸屏控制器，触摸屏控制器将接收到的信息转换成触点坐标，再送给 PLC，它同时接收 PLC 发来的命令，并加以执行。

**2. 触摸屏的分类**

触摸屏主要有电阻式触摸屏、电容式触摸屏、红外线式触摸屏和表面声波触摸屏等。

### 10.1.2　触摸屏的通信连接

触摸屏的图形界面是在计算机的专用软件（如 WinCC flexible）上设计和编译的，需要通过通信电缆下载到触摸屏；触摸屏要与 PLC 交换数据，它们之间也需要通信电缆。

**1. 计算机与西门子触摸屏之间的通信连接**

通常台式计算机上至少有一个 RS－232C 接口，西门子触摸屏有一个 RS－422/485 接口，个人计算机与触摸屏通过这两个接口进行通信，通常采用 PPI 通信（也可以采用 MPI、PROFIBUS 通信等），市场上有专门的 PC/PPI 电缆（此电缆也用于计算机与 S7－200 系列 PLC 的通信）出售，不必自己制作。当然也有 USB 接口形式的 PC/PPI 电缆出售，效果完全相同。如果读者使用的计算机上没有 RS－232C 接口，手头又没有 USB 接口形式的 PC/PPI 电缆，可以在市场上购置一个 USB－RS232C 转换器，但要注意使用 USB－RS232C 转换器前必须安装驱动程序。计算机与西门子触摸屏之间的联机最为简便的方案就是使用 PC/PPI 电缆。计算机与触摸屏通信连接如图 10-1 所示。

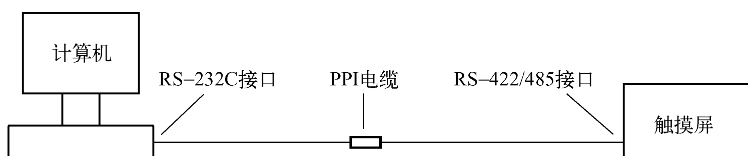



图 10-1　计算机与触摸屏通信连接

计算机与西门子触摸屏之间的联机还有其他方式，例如在计算中安装一块通信卡，通信卡自带一根通信电缆，将两者连接即可。如西门子的 CP6511 通信卡是 PCI 卡，安装在计算机主板的 PCI 插槽中，可以提供 PPI、MPI 和 PROFIBUS 等通信方式。有以太网口的触摸屏，计算机还可通过以太网向触摸屏下载程序。

**2. 触摸屏与 PLC 的通信连接**

西门子触摸屏有一个 RS－422/485 接口，西门子 S7－300/400 PLC 有一个编程口（MPI 口），两者互联实现通信采用的通信电缆接线如图 10-2 所示。

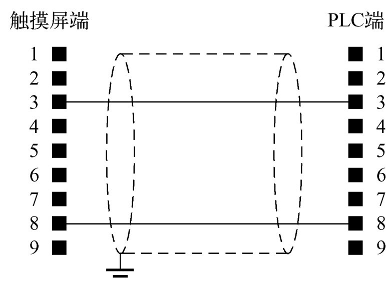



图 10-2　触摸屏与 S7－300/400 的通信连接

## 10.2　WinCC flexible 软件简介

### 10.2.1　认识 WinCC flexible 软件

**1. 初识 WinCC flexible 软件**

西门子的触摸屏过去用 ProTool 组态，WinCC flexible 是在 ProTool 组态软件的基础上发展而来的，多种语言使它可以在全球通用，因而被广泛认可。WinCC flexible 综合了 WinCC 的开放性和可扩展性，以及 ProTool 的易用性。

WinCC flexible 是西门子的全集成自动化的重要组成部分，西门子的全集成自动化（TIA）是指控制系统使用统一的通信协议、统一的数据库和统一的编程工具。WinCC flexible 可以与 STEP 7 V5.5、SCOUT 和 iMAP 集成在一起。

WinCC flexible 具有开放、简易的扩展功能，带有 Visual Basic 脚本功能，集成了 ActiveX 控件，可以将人机界面集成到互联网。WinCC flexible 易学易用、功能强大，提供了智能化的工具，例如图形导航和移动的图形化组态。在创建项目时，通过点击鼠标便可以生成触摸屏项目的基本框架结构。基于表格化的编辑器简化了对象（如变量、文本）的生成和编辑。通过图形化的配置，简化了复杂的配置任务。

**2. WinCC flexible 软件的组成**

WinCC flexible 软件包含三个组件：WinCC flexible 工程系统、WinCC flexible 运行系统和 WinCC flexible 选件。

（1）WinCC flexible 工程系统

WinCC flexible 工程系统用于开发组态运行与操作面板的监控管理系统，分为如下四个版本：

1）WinCC flexible 微型版，用于组态微型面板（Micro Panel）。

2）WinCC flexible 压缩版，用于组态 70 和 170 系列面板。

3）WinCC flexible 标准版，用于组态 270 和 370 系列面板。

4）WinCC flexible 高级版，用于组态基于 PC 面板和运行系统。

（2）WinCC flexible 运行系统

WinCC flexible 运行系统（WinCC flexible Runtime）用于过程可视化与自动化系统的通信，操作自动化生产过程，处理维护报警信息和各种数据。在 Windows 平台上运行用户组态程序，还可以在组态计算机上模拟、测试用户编译完成的组态文件。

（3）WinCC flexible 选件

WinCC flexible 选件包含 ChangeControl、Archive、Audit、Sm@ rtAccess、Sm@ rtServices、OPC 和 ProAgent 等。

### 10.2.2　安装 WinCC flexible 软件

**1. 安装 WinCC flexible 软件的条件**

WinCC flexible 有多种语言版本，目前最新版本是 WinCC flexible 2008 SP4，相对安装西门子的其他软件，安装 WinCC flexible 软件是比较费时间的。WinCC flexible 软件对计算机的硬件要求较高，推荐安装条件如下：

① 操作系统：Windows XP Professional 或者 Windows 7（专业版或者旗舰版，32 位和 64 位均可）。

② 主内存：1.5 GB 或者更大，推荐 2 GB 或者更大。

③ 硬盘空间：2 GB 或者更大。

④ CPU：Pentium 4 或更强。

⑤ Internet 浏览器：Microsoft Internet Explorer V6.0 SP1。

⑥ PDF 阅览器：Adobe Acrobat 5.0 或者更高。

以上安装要求是最低要求，作者推荐 i5 以上 CPU 和 2 GB 以上 RAM，这样运行程序会比较流畅，否则会有“卡机”现象。

**2. 安装 WinCC flexible**

安装 WinCC flexible 由读者参照说明书自行完成。但安装时有如下几个注意事项：

1）安装时最好将监控和杀毒软件关闭，而且源文件和安装目录最好是英文，否则可能不能安装该软件。

2）较早的 WinCC flexible 版本（2007 及以前的版本）与西门子的部分软件冲突，因此建议安装新版的 WinCC flexible。

3）操作系统如是 Windows XP，应选择专业版或者企业版；操作系统如是 Windows 7，应选择专业版或者旗舰版。不能使用家庭版的操作系统。

## 10.3　变量组态

### 10.3.1　变量类型

触摸屏中使用的变量类型和选用的控制器的变量是一致的，例如读者若选用西门子的

S7 - 300/400 系列 PLC，那么触摸屏中使用的变量类型就和 S7 - 300/400 系列 PLC 的变量类型基本一致。

### 10.3.2 函数

西门子的人机界面有很多函数，可分为记录函数、用户管理函数、画面函数、位处理函数、打印函数、设置函数、报警函数、配方函数、系统函数、键盘函数、用于画面对象的热键函数和其他函数。通常越高档的人机界面函数越丰富，使用越方便。以下介绍几个常用的函数。

**1. 位函数**

（1） InvertBit

其作用是对给定的“BOOL”型变量的值取反。如果变量现有值为 1（真），它将被设置为 0（假）；如果变量现有值为 0（假），它将被设置为 1（真）。

（2） ResetBit

将“BOOL”型变量的值设置为 0（假）。

（3） SetBit

将“BOOL”型变量的值设置为 1（真）。

（4） SetBitWhileKeyPressed

只要用户按下已组态的键，给定变量中的位即设置为 1（真）。在改变了给定位之后，系统函数将整个变量传送回 PLC。但是并不检查变量中的其他位是否同时改变。在变量被传送回 PLC 之前，操作员和 PLC 只能读该变量。仅应使用该函数访问“BOOL”类型的变量，以避免发生同时访问相同变量的问题。

**2. 计算函数**

（1） IncreaseValue

将给定值添加到变量值上，用方程表示为 $X = X + a$。

系统函数使用同一变量作为输入和输出值。当该系统函数用于转换数值时，必须使用帮助变量。可使用系统函数“SetValue”将变量值分配给帮助变量。

如果在报警事件上组态了函数且变量未在当前画面中使用，则无法确保在 PLC 中使用实际的变量值。通过设置“连续循环”采集模式可以改善这种情况。

（2） SetValue

将新值赋给给定的变量。该系统函数可用于根据变量类型分配字符串和数字。

**3. 画面函数**

（1） ActivateScreen

将画面切换到指定的画面。使用“ActivateScreenByNumber”系统函数可以从根画面切换到永久性窗口，反之亦然。

（2） ActivatePreviousScreen

将画面切换到在当前画面之前激活的画面。如果先前没有激活任何画面，则画面切换不执行。最近调用的 10 个画面被保存。当切换到不再保存的画面时，会输出一条系统消息。

**4. 用户管理**

（1） Logoff

在 HMI 设备上注销当前用户。

（2） Logon

在 HMI 设备上登录当前用户。

（3） GetUserName

在给定的变量中写入当前登录到 HMI 设备用户的用户名。如果给出的变量具有控制连接，则用户名在 PLC 上也可用。该系统函数将使诸如执行某些功能与用户有关的版本成为可能。

（4） GetPassword

在给定的变量中写入当前登录到 HMI 设备的用户的口令。确保给定变量的值未显示在项目中的其他位置。

**5. 报警函数**

（1） EditAlarm

为选择的所有报警触发“编辑”事件。如果要编辑的报警尚未被确认，则在调用该系统函数时自动确认。

（2） ShowAlarmWindow

隐藏或显示 HMI 设备上的报警窗口。

（3） ClearAlarmBuffer

删除 HMI 设备报警缓冲区中的报警。尚未确认的报警也被删除。

（4） AcknowledgeAlarm

确认选择的所有报警。该系统函数用于 HMI 设备没有 ACK 键时或报警屏幕的集成键不能使用时。

## 10.4 画面组态

### 10.4.1 按钮组态

按钮的主要功能是在点击它的时候执行事先组态好的系统函数，使用按钮可以完成丰富多彩的任务。

**1. 用按钮增减变量值**

新建一个项目。打开画面，选中“工具箱”中的“简单对象组”，将其中的“按钮”拖入到画面的工作区，选中按钮。在按钮的属性视图的“常规”对话框中，设置按钮模式为“文本”。设置“OFF”状态文本为“+10”，如图 10-3 所示。如果未选中“ON”复选框，按钮在按下时和弹起时的文本相同。如果选中它，按钮在按下时和弹起时，文本的设置可以不相同。

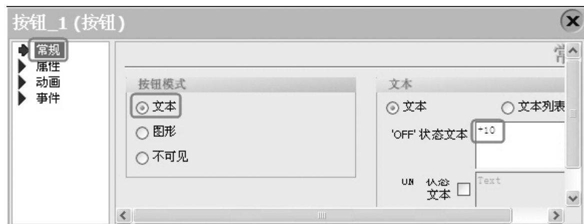



图 10-3　按钮的属性组态

打开按钮属性视图的“事件”内的“单击”对话框，如图 10-4 所示，单击按钮时，执行系统函数列表“计算”文件夹中的系统函数“IncreaseValue”，被增加的整型变量是“X”，增加值是 10。




图 10-4　按钮触发事件组态（1）

在按钮的上方生成一个 3 位整数的输出 IO 域，如图 10-5 所示，连接变量的类型为“X”。当按下工具栏的“仿真”按钮，处于模拟运行状态，开始离线模拟运行。每单击一次按钮，IO 域中的数值增加 10。

**2. 用按钮设定变量的值**

新建一个项目。打开画面，选中“工具箱”中的“简单对象组”，将其中的“按钮”拖入到画面的工作区，选中按钮。在按钮的属性视图的“常规”对话框中，设置按钮模式为“文本”。设置“OFF”状态为“1”，方法与以上例子相同。

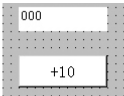



图 10-5　按钮和 IO 域的画面

打开按钮属性视图“事件”内的“单击”对话框，如图 10-6 所示，单击按钮时执行系统函数列表“计算”文件夹中的系统函数“SetValue”，被设置的整型变量是“Y”，Y 数值变成 20。

在按钮的上方生成一个 3 位整数的输出 IO 域，如图 10-7 所示，连接变量的类型为“Y”。当按下工具栏的“”，处于模拟运行状态，开始离线模拟运行。每次单击按钮，IO 域中的数值均为 20。




图 10-6　按钮触发事件组态（2）

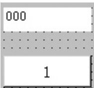



图 10-7　按钮和 IO 域的画面

## 10.4.2　IO 域组态

I 是输入（Input）的简称，O 是输出（Output）的简称，输入域和输出域统称 IO 域。IO 域应用在触摸屏中比较常见。

**1. IO 域的分类**

1）输入域：用于操作员输入要传送到 PLC 的数字、字母或符号，将输入的数值保存到变量中。

2）输出域：只显示变量数据。

3）输入/输出域：同时具有输入和输出功能，操作员可以用它来修改变量的数值，并将修改后的数值显示出来。

**2. IO 域的组态**

先建立“连接 1”，在变量表中建立整型变量（Int）“MW0”“MW2”和“MW4”，如图 10-8 所示。再生成和打开“IO 域”画面，选中工具箱中的“简单对象”，将“IO 域”对象拖到画面编辑器的工作区。在画面上建立 3 个 IO 域对象，如图 10-9 所示。分别在 3 个 IO 域的属性视图的“常规”对话框中，设置模式为“输入”“输出”和“输入/输出”，如图 10-10 所示。

| 名称 | 连接 | 数据类型 | 地址 | 数组计数 | 采集周期 |
|---|---|---|---|---|---|
| MW0 | 连接_1 | Int | MW 0 | 1 | 100 ms |
| MW2 | 连接_1 | Int | MW 2 | 1 | 100 ms |
| MW4 | 连接_1 | Int | MW 4 | 1 | 100 ms |

图 10-8　新建变量

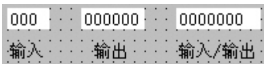



图 10-9　IO 域组态




图 10-10　输入域的常规属性组态

输入域显示 3 为整数，为此组态“移动小数点”（小数部分的位数），“格式样式”为“999”，表示整数为 3 位，变量的更新周期为 100 ms。

## 10.4.3　开关组态

开关是一种用于布尔（BOOL）变量输入、输出的对象，它有两项基本功能：一是用图形或者文本显示布尔变量的值（0 或者 1）；二是点击开关时，切换连接的布尔变量的状态，如果原来是 1 则变为 0，如果原来是 0 则变为 1，这一功能集成在对象中，不需要用户组态，发生“单击”事件时执行函数。

**1. 切换模式的开关组态**

将“工具箱”中的“简单对象”组中的“开关”拖放到画面的编辑器中。切换模式开关如图 10-11 所示，方框的上部是文字标签，下部是带滑块的推拉式开关，中间是打开和关闭对应的文本。在开关属性视图的“常规”对话框中，选择开关模式为“切换”，如图 10-12 所示，开关与变量“启停”连接，将标签“Switch”改为“变频器”，“ON”和“OFF”状态的文本由“0”和“1”改为“启”和“停”。

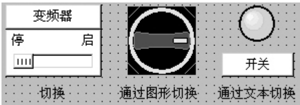



图 10-11　开关画面




图 10-12　开关常规属性

当按下工具栏的“起动仿真”按钮时，处于模拟运行状态，开始离线模拟运行。

**2. 通过图形切换模式的开关组态**

WinCC flexible 的图形库中有大量的控件可供用户使用。在工具栏的“图形”组中打开 WinCC flexible 图形文件夹→Symbol Factory Graphics→Symbol Factory Type Color→3 - D Pushbuttons Ets，如图 10-13 所示。读者可以在安装 WinCC flexible 的文件目录找到图形文件。

将工具箱的“简单对象”中的“开关”拖放到画面编辑器中，如图 10-11 所示。在常规视图对话框中，将组态开关的类型设置成“通过图形切换”，过程变量与“启停”连接，选择“1”处，如图 10-14 所示。这样两个开关就组态完成。

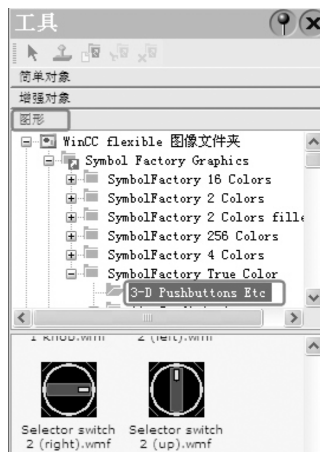



图 10-13　图形库路径




图 10-14　图形切换开关组态

**3. 通过文本切换模式的开关组态**

将工具箱的“简单对象”中的“开关”拖放到画面编辑器中，如图 10-11 所示。在常规视图的对话框中，将组态开关的类型设置成“通过文本切换”，过程变量与“启停”连接，将“ON”状态设置为“起动”，将“OFF”状态设置为“停止”，如图 10-15 所示。当按下工具栏的按钮时，处于模拟运行状态，开始离线模拟运行。文本在起动和停止之间切换时，灯随之亮或灭。



图 10-15　开关常规属性

## 10.4.4　图形输入/输出对象组态

### 1. 棒图的组态

棒图以带刻度的棒图形式表示控制器的值。通过 HMI 设备，操作员可以立即看到当前值与组态的限制值相差多少或者是否已经达到参考值。棒图可以显示诸如填充量（水池的水量、温度数值）或批处理数量等值。

在变量表中创建整型（INT）变量“温度”，只要单击工具栏中“简单对象”中的“棒图”，用鼠标拖动即可得到如图 10-16 所示的棒图（图的左边是拖动过程中，图的右侧是拖动完成的棒图）。

在属性的“常规”对话框中，设置棒图连接的整型变量为“温度”，如图 10-17 所示，温度的最大值和最小值分别是 100℃和 0℃，这两个数值是可以修改的。当温度变化时，棒图画面中的填充色随之变化，就像温度计一样。

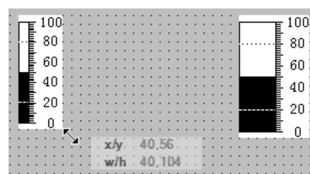


图 10-16　棒图画面



图 10-17　棒图常规属性组态

### 2. 量表的组态

量表是一种动态显示对象。量表通过指针显示模拟量数值。例如，通过 HMI 设备操作员一眼就能看出锅炉压力是否处于正常范围之内。以下是量表的组态方法。

生成和打开“量表”画面，如图 10-18 所示，将工具箱的“增强对象”中的“量表”图标拖到画面中，在量表的属性视图的“常规”对话框中，可以设置显示物理量的单位，本例为“kM”，“标签”在量表圆形表盘的下部显示，可以选择是否显示峰值（一条沿半径方向的红线，本例在刻度 0 处），如图 10-19 所示。读者还可以自定义背景图形和表盘图形。

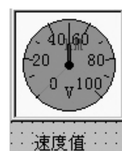


图 10-18　量表画面组态

量表的“刻度”属性如图 10-20 所示，读者可以设置刻度的最大值和最小值、表盘圆弧的起始角度和终止角度。“分度”是指相邻的两个刻度之间的数值，可以修改。量表可以用几种不同的颜色来表示正常范围、警告范围和危险范围。




图 10-19　量表常规属性组态




图 10-20　量表刻度属性组态

量表除了有“常规”属性和“刻度”属性外，还有“外观”属性（主要设置背景颜色、钟表颜色和表盘样式等）、“文字”属性（主要设置字体大小和颜色等）以及“布局”属性（主要是表盘画面的位置和尺寸），这些属性都比较简单，在此不再赘述。

**3. 滚动条组态**

滚动条控件用于操作员输入和监控变量的数字值。用来显示数字值时，滚动条的滑块位置用来指示控件输出的过程值。操作员通过改变滑块的位置来输入数字值。滚动条的画面组态如图 10-21 所示。

将工具栏的“增强对象”组中的“滚动条”图标拖入图形画面中。滚动条属性视图的“常规”窗口与棒图的“常规”窗口相同。

在“图样”窗口的“标签”中，可以输入滚动条的物理量的单位，它显示在滚动条的上方，如图 10-22 所示。




图 10-21　滚动条控件画面组态




图 10-22　滚动条的图样组态

在“边框”窗口中，可以用图形形象地说明各个参数的含义，如图 10-23 所示。属性中的其他设置由读者自行设定。

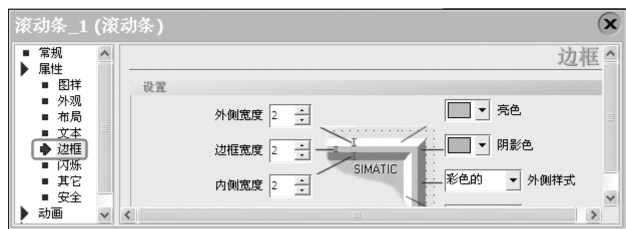



图 10-23　滚动条的边框组态

**4. 离线模拟运行**

单击 WinCC flexible 工具栏的按钮，起动模拟运行器的运行系统，开始离线模拟运行。首次运行时，模拟运行器是一张空白表格，如图 10-24 所示。可以看到滑块在最下面，将变量选定为“温度”，在设定值中输入一个数值（本例为 58），按计算机的回车键后，弹出如图 10-25 所示的界面。可以看到滑块指向 58℃，即当前值为 58℃。模拟的类型还有 Sine（正弦）、随机、位移、增量和减量。



图 10-24　空白模拟器



图 10-25　运行模拟器

## 10.4.5　时钟和日期的组态

生成和打开“日期时间”的画面，如图 10-26 所示，将工具箱中的“简单视图”组中的“日期时间域”图标和“增强视图”中的“时钟”拖至画面中。有的触摸屏没有时钟对象（如 TP－177B）。

选中“日期时间”域，做如图 10-27 所示的设置，当按下工具栏的按钮，处于模拟运行状态，开始离线模拟运行，日期时间域中显示系统的当前时间，并不需要设置任何参数。

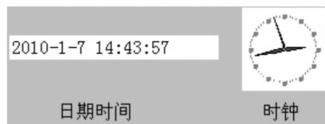



图 10-26　“日期时间”的画面组态




图 10-27 “日期时间”的常规组态

## 10.4.6 间接寻址和符号 IO 域组态

**1. 双状态符号 IO 域**

双状态符号 IO 域用位变量切换两个不同的文本。如图 10-28 所示，符号 IO 域与位变量电动机的“开/关”关联，用于显示系统的工作模式。




图 10-28 “符号 IO 域”的常规组态

要在运行系统中控制电动机，操作员从文本列表中选择文本“电动机关”或“电动机开”。根据选择，随后将起动或关闭电动机。符号 IO 域显示电动机的相应状态。

**2. 多状态符号 IO 域**

多状态符号域相对要麻烦些，以下用一个例子来说明多状态符号 IO 域的用法。

某设备有三个测温点，用 HMI 的多状态符号域来显示这三个点的温度。

新建一个项目，并新建 3 个过程变量“温度 1”“温度 2”和“温度 3”以及内部变量“温度值”和“温度指针”，如图 10-29 所示。

| 名称 | 连接 | 数据类型 | 地址 | 数组计数 | 采集周期 | 注释 |
|---|---|---|---|---|---|---|
| 启停 | 连接_1 | Bool | M 0.0 | 1 | 1 s | |
| 温度1 | 连接_1 | Int | MW 0 | 1 | 1 s | |
| 温度2 | 连接_1 | Int | MW 2 | 1 | 1 s | |
| 温度3 | 连接_1 | Int | MW 4 | 1 | 1 s | |
| 温度值 | <内部变量> | Int | <没有地址> | 1 | 1 s | |
| 温度指针 | <内部变量> | Int | <没有地址> | 1 | 1 s | |

图 10-29 新建参数

在工具栏中将“符号 IO 域”拖入画面，并选中“符号 IO 域”，在“常规”中新建名称为“温度值”的文本列表，它的三个项目分别为“温度 1”“温度 2”和“温度 3”，如图 10-30 所示，三个温度值对应三个索引。

图 10-30　文本列表

选中“常规”→“变量”→“温度指针”，再选定“温度值”，单击“2”处的图标，如图 10-31 所示。此时弹出如图 10-32 所示的界面，选中“指针化”，再选择“温度指针”，接着在变量栏中选择“温度 1”“温度 2”和“温度 3”，最后单击“确定”按钮即可。

图 10-31　变量的间接地址（1）

图 10-32　变量的间接地址（2）

在画面中，将输出域与变量“温度值”连接，输出域的常规组态如图 10-33 所示。

图 10-33　输出域的常规组态

单击 WinCC flexible 工具栏的按钮，起动模拟运行器的运行系统，开始离线模拟运行，如图 10-34 所示。把表格中的“温度 1”“温度 2”和“温度 3”设定为不同大小的数值，当在“温度值”中选择不同的温度选项，“温度显示”中对应显示不同的温度值，如本例中温度 3 的数值为 88。当温度测量点较多时（如 5 个或者更多），这种方法的好处就十分明显。




图 10-34　运行模拟器

## 10.4.7　图形 IO 域组态

前面的文本 IO 域可以用位变量实现文本的切换，而图形 IO 域和图形列表的功能是切换多幅图形，从而实现丰富多彩的动画效果。图形 IO 域有输入、输出、输入/输出和双状态四种。以下用一个例子（完成叶片转动效果的动画）来讲解图形 IO 域的使用方法。

新建一个 HMI 工程，并将工具栏“简单对象”中的“图形 IO 域”拖入画面，如图 10-35 所示。再在“变量”表中创建变量“电动机指针”，如图 10-36 所示。




图 10-35　运行模拟器

| 名称 | 连接 | 数据... | 地址 | 数组计数 | 采集... | 注释 |
|---|---|---|---|---|---|---|
| 电动机指针 | 连接_1 | Int | MW 0 | 1 | 1 s | |

图 10-36　新建变量

在其他的绘图工具（如 Visio 或者 AutoCAD 等）中绘制叶片转动时的三个状态，并将其保存为“图形_1. gif”“图形_2. gif”和“图形_3. gif”，存放到计算机的某个空间上。这三个图形外观如图 10-37 所示，当在图形域中不断按顺序装载这三幅图片时，就产生了动画效果（类似电影的原理）。

在图形列表中创建变量“电动机”，如图 10-37 所示的“1”处，再单击“2”处（单击之前，并没有“图形_1”字样），将“图形_1”装载到位，装载图形时，读者要明确事先将“图形_1. gif”“图形_2. gif”和“图形_3. gif”存放在计算机中确切的位置。

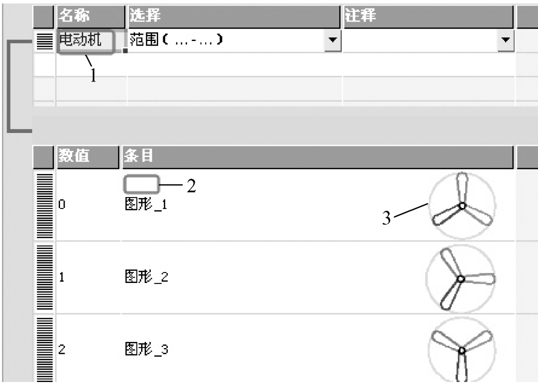



图 10-37　图形列表

在画面中选中“图形 IO 域”，设置如图 10-38 所示。




图 10-38　图形 IO 域常规属性组态

在画面中拖入两个按钮，一个按钮命名为“加 1”，组态时，事件为“单击”，函数为“IncreaseValue”，变量为“电机指针”；一个按钮命名为“减 1”，组态时，事件为“单击”，函数为“DecreaseValue”，变量为“电动机指针”，画面如图 10-39 所示。这样当单击“加 1”按钮时，换一幅画面，产生动画效果。

单击 WinCC flexible 工具栏的按钮，起动模拟运行器的运行系统，开始离线模拟运行。当单击“加 1”按钮一次，换一幅画面，产生动画效果是逆时针旋转。当单击“减 1”按钮一次，换一幅画面，产生动画效果是顺时针旋转。如图 10-40 所示的位置，是按 2 次“加 1”按钮时的图面。

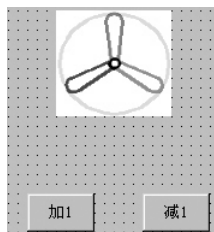



图 10-39　图形 IO 域画面

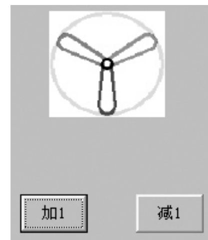



图 10-40　图形 IO 域运行画面

**【关键点】** 当“电动机指针”的数值大于或等于 3 时，会出错。这是因为，本例图片的索引值最大为 2。同理，当“电动机指针”的数值小于或等于 -1 时，也会出错。

## 10.4.8　画面的切换

双击“项目管理器”下的“新建 画面”，新建“画面 2”和“画面 3”，“画面 1”不需要新建，新建项目时就存在，如图 10-41 所示。选中“画面 1”，拖入三个按钮，分别命名为“画面 2”“画面 3”和“停止系统运行”。

在画面 1 中，选中按钮“画面 2”，再选中事件中的“单击”，选中函数中的“ActivateScreen”（激活画面），选择激活画面函数的参数为“画面 2”，如图 10-42 所示。这样做的目的是当单击按钮“画面 2”时，从当前画面（画面 1）转到画面 2。如图 10-43 所示，其含义是退出运行在 HMI 设备上的项目。

选中“画面 3”，拖入三个按钮，分别命名为“画面 2 ”“前一画面”和“返回父画面”，如图 10-44 所示。

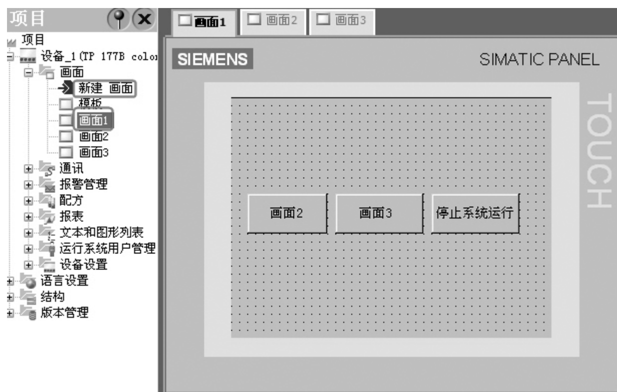



图 10-41　新建画面




图 10-42　按钮“单击”事件组态（1）




图 10-43　按钮“单击”事件组态（2）

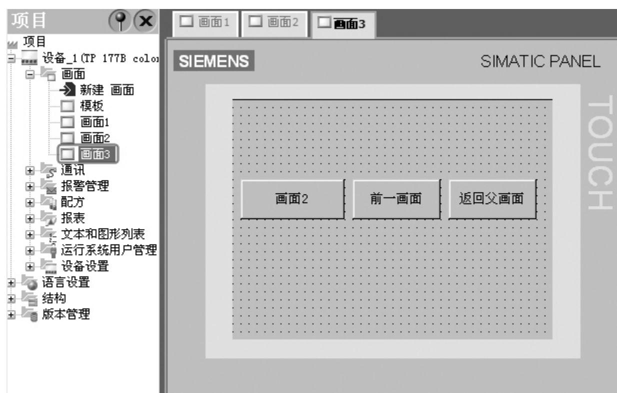



图 10-44　画面 3

在如图 10-44 中，选中按钮“前一画面”，再选中事件中的“单击”，选中函数中的“ActivatePreviousScreen”（激活前一画面），无参数，如图 10-45 所示。这样做的目的是当单击按钮“前一画面”时，从当前画面（画面 3）转到前一个画面。

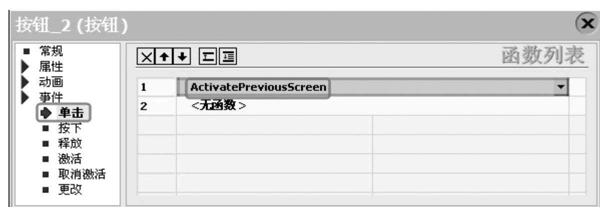



图 10-45　按钮“单击”事件组态（3）

在如图 10-44 中，选中按钮“返回父画面”，再选中事件中的“单击”，选中函数中的“ActivateParentScreen”（激活父画面），无参数，如图 10-46 所示。这样做的目的是当单击按钮“返回父画面”时，从当前画面（画面 3）转到返回父画面。




图 10-46　按钮“单击”事件组态（4）

## 10.5　用户管理

### 10.5.1　用户管理的基本概念

**1. 应用领域**

控制系统在运行时，有时需要修改某些重要的参数，例如温度、压力、时间，以及 PID 控制器的参数值等。很显然这些重要的参数修改只允许某些指定的人员才能操作，以防止某些未授权的人员对这些重要的数据的访问和修改，而造成某些不必要的损失。通常操作工只能访问指定输入域和功能键，权限最低，而调试工程师则可以不受限制地访问所有的变量，其权限较高。

**2. 用户组和用户**

用户管理主要涉及两类对象：用户组和用户。

用户组主要设置某一类用户的组具有的特定权限。用户属于某一个特定的用户组。一个用户只能分配给一个用户组。

在用户管理中，访问权限不能直接分配给用户，而是分配给特定的用户组。某一特定用户被分配到特定的用户组以获得权限。这样，对待特定用户的管理就和对权限的组态分离开来了，方便编程人员组态。

### 10.5.2　用户管理编辑器

**1. 添加用户组**

在“项目视图”中展开“运行系统用户管理”条目，双击“组”选项，创建新的项目组，如图 10-47 所示。系统默认情况下，已经预定义操作员组和管理员组两类用户组。而“经理”“工程师”和“组长”用户组则是添加的用户组。新创建的组的名称一般从“组_

1”开始编号，这个名称不符合实际，因此要重命名，只要选中该条目，并在常规属性下的名称中输入所需的名称即可，如图 10-48 所示。



图 10-47　添加用户组



图 10-48　更改用户组名称

**2. 添加用户**

在“项目视图”中展开“运行系统用户管理”条目，双击“用户”选项，创建新的用户，设置口令，并将其分配给特定的用户组，如图 10-49 所示。选中如图 10-50 中的用户名（如 Xixiaowang），并在常规属性下的名称中输入所需的用户名称，在口令中输入口令（本例设置的口令为 123），如图 10-50 所示。



图 10-49　添加用户



图 10-50　更改用户名称、设置密码

## 10.5.3　访问保护

在工程系统中创建用户和用户组，并为其分配权限，之后可以为画面中的对象组态权

限。访问保护用于对控制数据和函数的访问。将组态传送到 HMI 后，所有组态了权限的画面对象均会得到保护，以避免在运行时受到未经授权的访问。以下用一个例子介绍访问保护问题。触摸屏工程有两个画面，主画面如图 10-51 所示，参数设定画面需要有权限才能进入。

先将工具栏“增强对象”中的“用户视图”拖入主画面，再向主画面中拖入三个按钮，如图 10-51 所示。将第一个按钮命名为“用户登录”，与按钮相连的事件为“单击”，函数为“ShowLogonDialog”；将第二个按钮命名为“用户注销”，与按钮相连的事件为“单击”，函数为“Logoff”；将第三个按钮命名为“参数设定”，与按钮相连的事件为“单击”，函数为“ActivateScreen”，画面为 2，并选中属性条目下的“安全”，将权限中的选项定为“管理”，如图 10-52 所示，这样要访问“参数设定”画面，至少需要“管理”权限。




图 10-51　运行时用户视图




图 10-52　安全属性组态

## 10.6　创建一个简单的触摸屏的工程

利用一台西门子 TP－270 触摸屏控制一台西门子 CPU314C－2DP 上一盏灯的开/关，并在触摸屏上显示灯的明暗状态。创建项目过程如下：

1）PC 与 HMI 通信连接。用一根 PC/PPI 编程电缆，将个人计算机与触摸屏相连，参考图 10-1。

2）新建项目。起动计算机中的 WinCC flexible 软件，初始界面如图 10-53 所示。

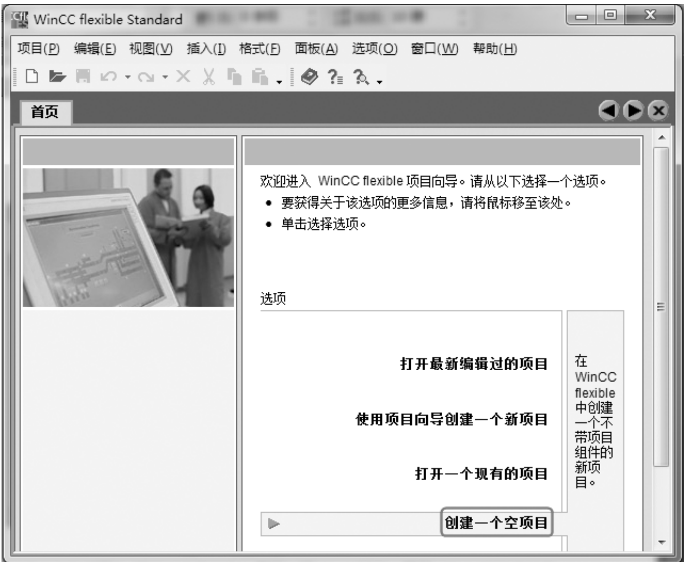



图 10-53　初始界面

双击“创建一个空项目”，弹出“设备选择”界面，如图 10-54 所示。展开“Panel”选中设备“TP 270 6””，再单击“确定”按钮。




图 10-54 “设备选择”界面

3）新建连接。展开左边“项目窗口”中的“通讯”项，双击“连接”项，再双击右边“名称”下面的空白，自动生成“连接_1”，接着在“通信驱动程序”中选择 SIMATIC S7 300/400。单击保存按钮，将工程保存一个名称，如默认的“项目 1. hmi”。连接编辑器（1）如图 10-55 所示。

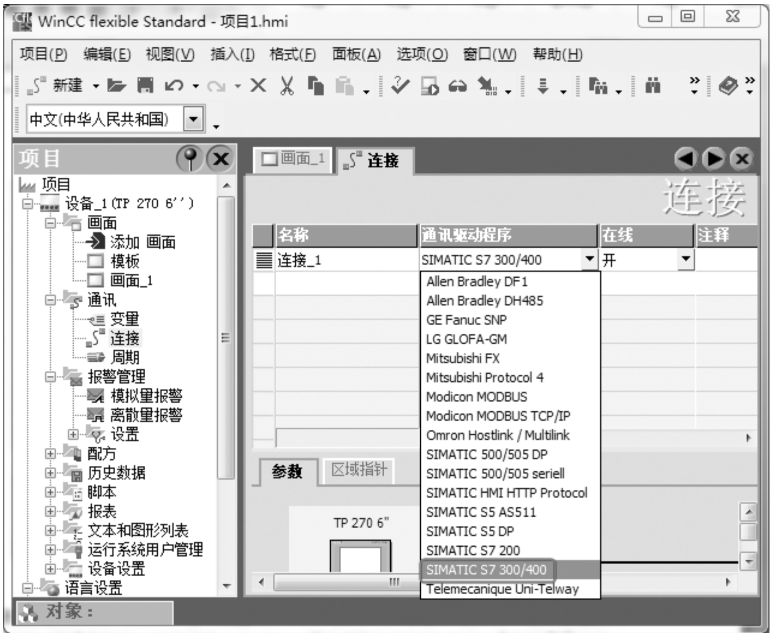



图 10-55 连接编辑器（1）

选择通信的波特率为“187500”，这个数据与 S7－300 系列 PLC 的默认波特率一致，在选择网络配置文为“MPI”，即通信协议，由于只有一台 PLC，因此其默认地址为“2”，通常不需要改变，但注意这个地址必须与 PLC 的地址一致。连接编辑器（2）如图 10－56 所示。



图 10-56　连接编辑器（2）

4）变量的生成与属性的设置。双击“变量”自动生成“变量编辑器”，如图 10-57 所示。双击“名称”下面的空白，自动生成变量，将“变量 1”重命名为“M00”，将“变数 2”重命名为“Q00”，将两者数据类型都改为“BOOL”，将“M00”的地址改为“M0.0”，将“Q00”的地址改为“Q0.0”，保存以上设置。

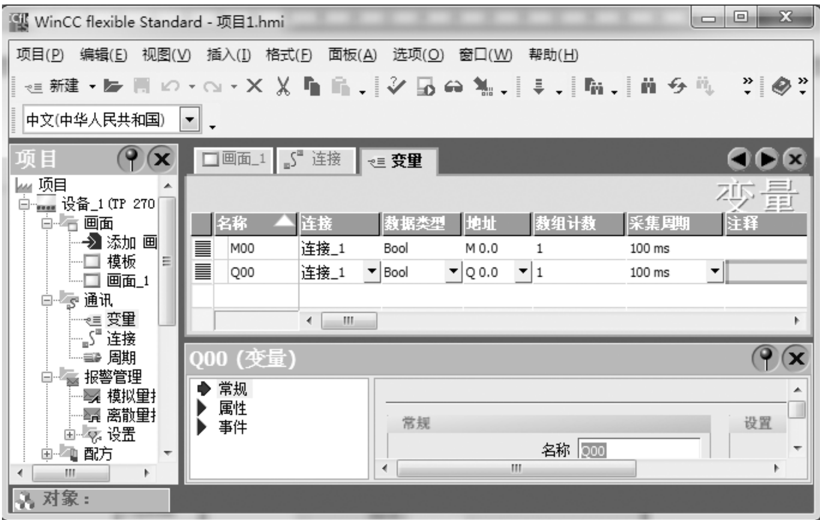


图 10-57　变量编辑器

5）创建画面。双击“项目窗口”下的“画面 1”，弹出画面编辑器，在左侧的“工具栏”中单击“按钮”，在画面编辑区拖出“按钮”，同样在“工具栏”中单击“圆”，在画面编辑区拖出“圆”，画面编辑器如图 10–58 所示。

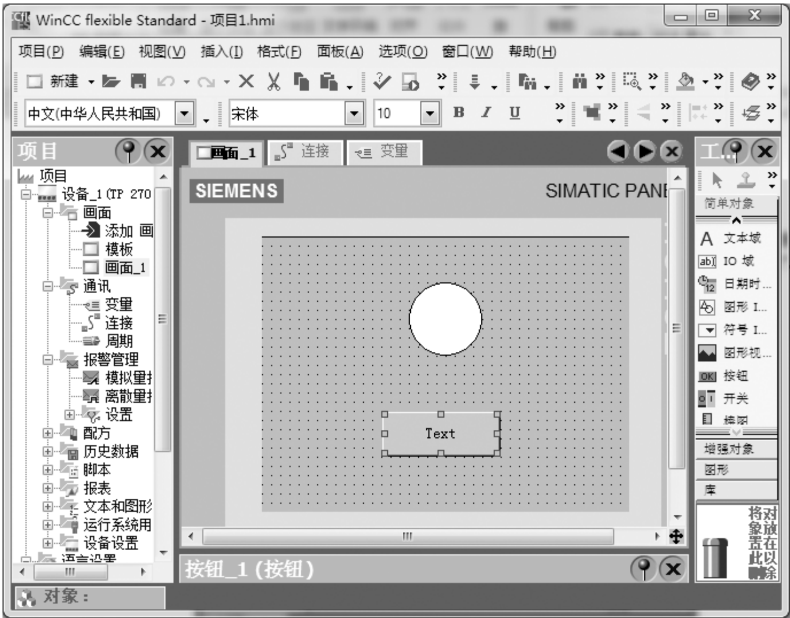



图 10–58　画面编辑器

6）图形对象与变量链接。

① 按钮的组态。选中“按钮”，再选中“属性窗口”中的“常规”，把“状态文本”中的“Text”改为“开关”。按钮组态（1）如图 10–59 所示。

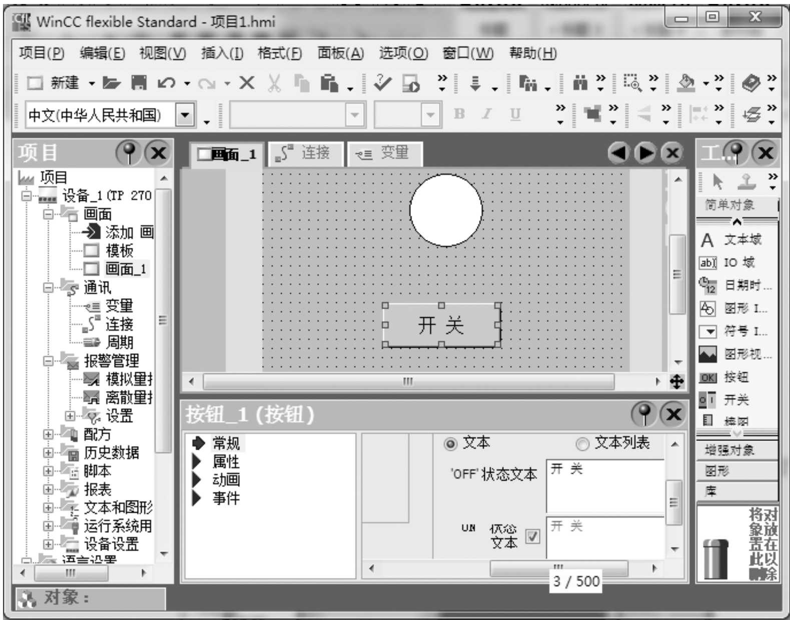



图 10–59　按钮组态（1）

② 选中“按钮”，再选中“属性窗口”中“事件”下的“单击”，再把函数选定为“编辑位”函数中的“InvertBit”。按钮组态（2）如图 10-60 所示。

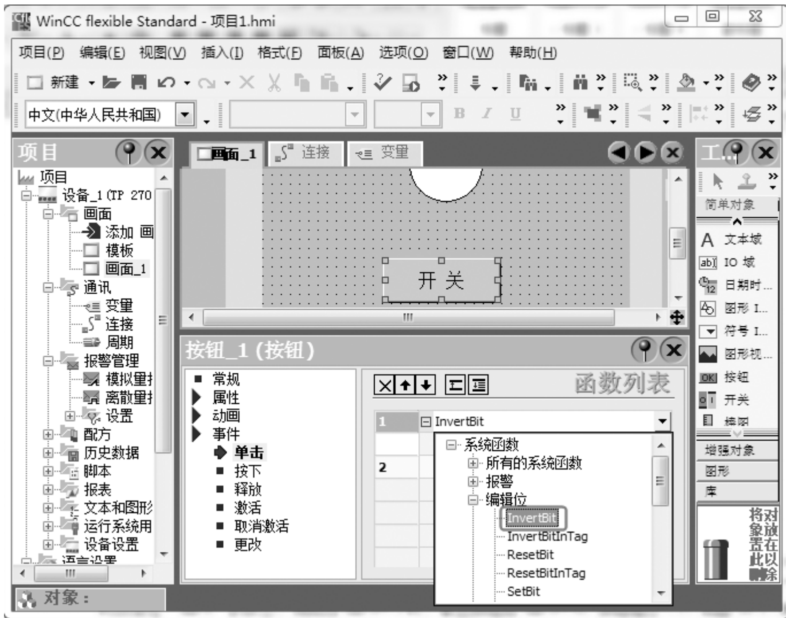


图 10-60　按钮组态（2）

③ 组态变量。建立了事件，但没有变量，仍然不能产生动作，因此在“变量”中，选定 M0.0，这样单击按钮“开关”就可以产生变量 M0.0 的变化。函数“InvertBit”的作用是每次单击“按钮”时改变位“M0.0”的状态，例如当 M0.0 = 1 时，单击按钮后M0.0 = 0。按钮组态（3）如图 10-61 所示。

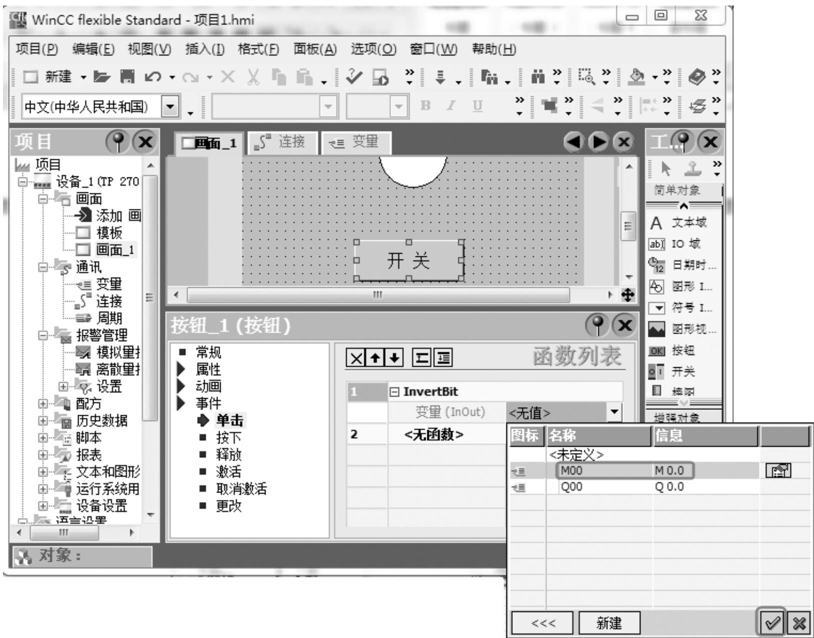


图 10-61　按钮组态（3）

④ 灯的组态。选中“圆”，将“属性窗口”→“属性”→“外观”中的“填充颜色”改为“红色”。灯的组态（1）如图 10-62 所示。

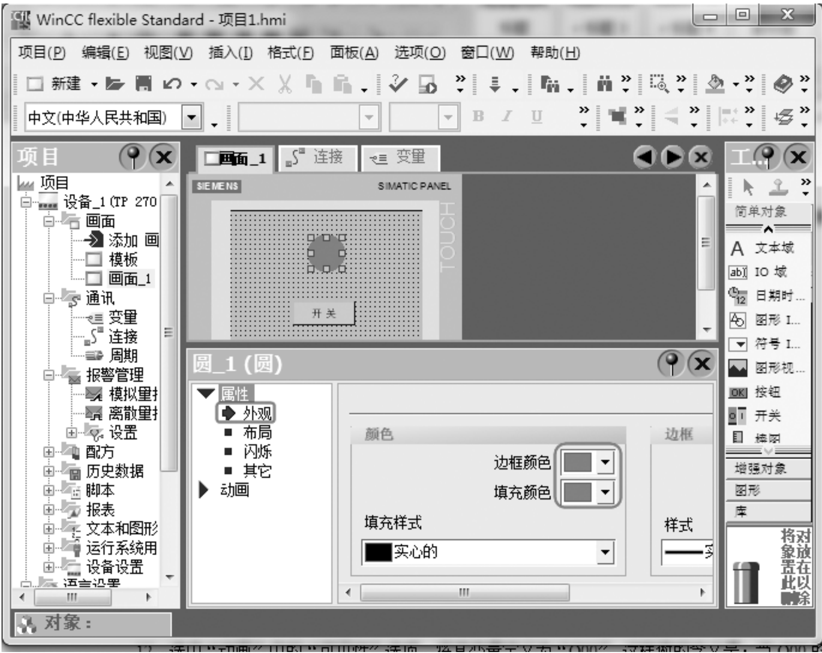


图 10-62　灯的组态（1）

⑤ 选中“动画”中的“可见性”选项，将其变量定义为“Q00”。这样做的含义是：当 Q00 的状态改变时，灯的可见性随之改变。灯的组态（2）如图 10-63 所示。

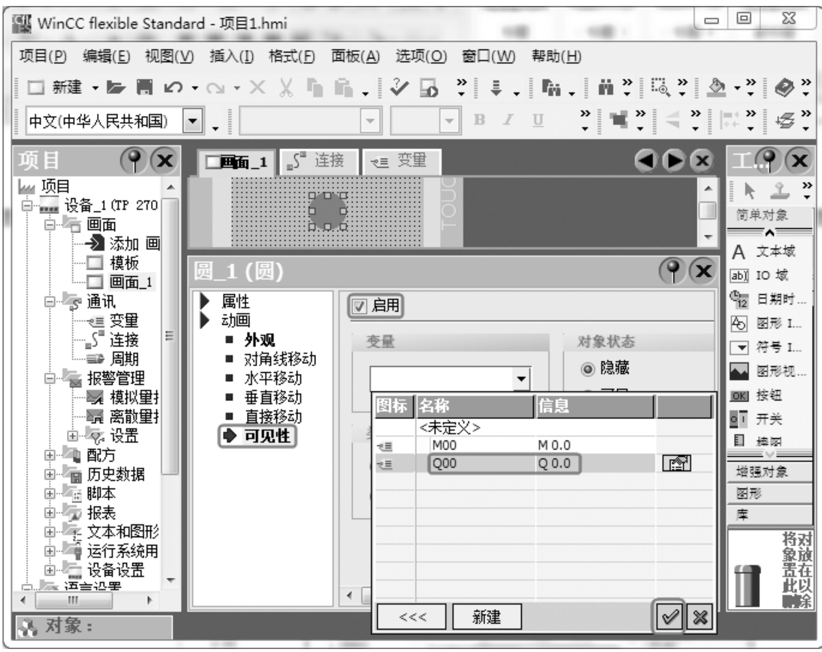


图 10-63　灯的组态（2）

⑥ 将对象的状态选定为“可见”。保存以上设置。灯的组态（3）如图 10-64 所示。

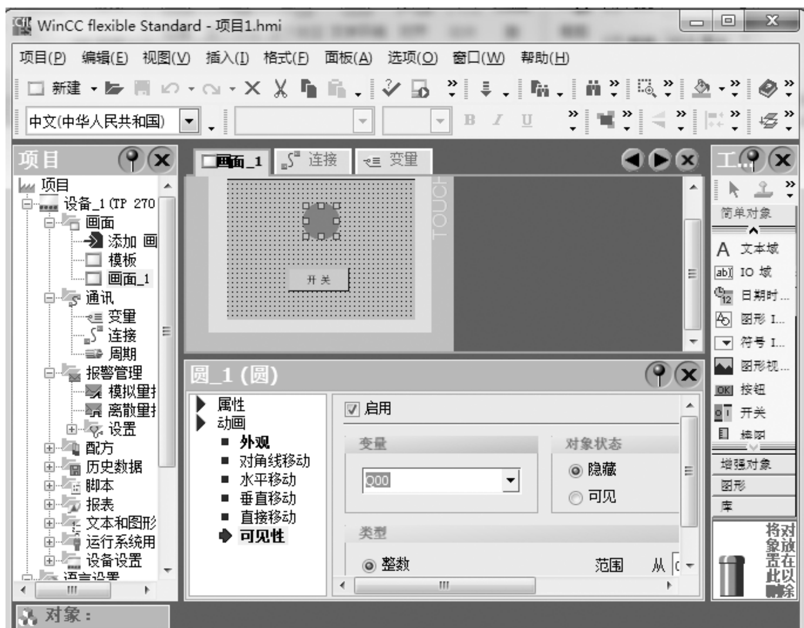



图 10-64　灯的组态（3）

7）下载项目。开启触摸屏的电源，触摸屏上出现初始界面如图 10-65 所示，用手触摸“Transfer”按钮，触摸屏准备与计算机通信。

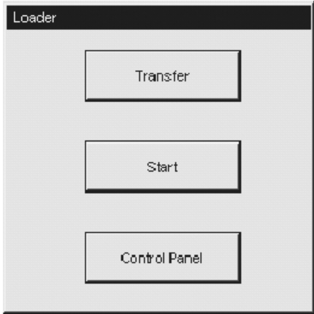



图 10-65　触摸屏初始界面

选择设备进行传送。在工具栏上单击“传送设置”按钮，弹出“选择设备进行传送”界面，如图 10-66 所示。本例中的“模式”选定为“串口”，选定端口为“COM1”。单击“传送”按钮，工程即可下载到触摸屏中。

计算机向触摸屏传送数据时的界面如图 10-67 所示。传送数据完成时，计算机上有“传送成功”的提示，若传送不成功也有相关提示。




图 10-66　“选择设备进行传送”界面




图 10-67　传送数据界面

再向 CPU 314C－2DP 中下载程序，如图 10-68 所示。




图 10-68　PLC 中的程序

参照图 10-2 制作电缆，并将 TP－270 触摸屏和 CPU 314C－2DP 连接在一起，应该在断电状态下连接通信电缆。

8）运行系统。接通触摸屏和 PLC 的电源，触摸屏上出现如图 10-65 的界面，用手触摸“Start”按钮，触摸屏中弹出运行界面，如图 10-69 所示。当用手触摸触摸屏上的“按钮”时，触摸屏上的灯变亮（白色），而且 PLC 的 Q0.0 闭合（PLC 上的 Q0.0 指示灯亮），当再次用手触摸触摸屏上的“按钮”时，触摸屏上的灯变暗（黑色），而且 PLC 的 Q0.0 断开（PLC 上的 Q0.0 指示灯灭）。

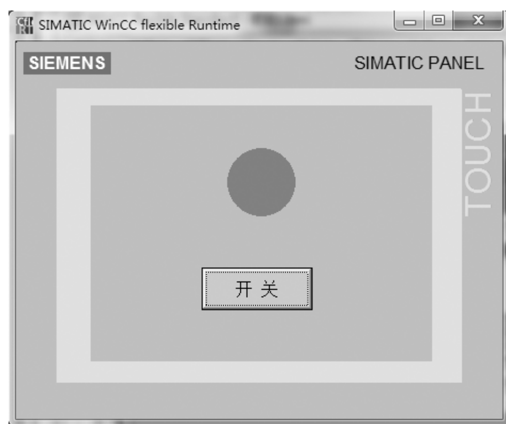



图 10-69　触摸屏运行界面

# 第 11 章　S7-300 PLC 在变频调速系统中的应用

本章介绍 MM440 变频器的基本使用方法、PLC 控制变频器多段频率给定、PLC 控制变频器模拟量频率给定、PROFIBUS 现场总线通信频率给定和 STARTER 软件的使用。

## 11.1　变频器的概述

### 11.1.1　变频器的发展历史

**1. 变频器之前的几件大事**

在讲解变频器的之前，有几件与之相关的大事不得不提，具体如下：

1）1831 年，英国的物理学家法拉第发现电磁感应原理，这使得人类使用电力成为可能。

2）1832 年，法国的皮克西研制出世界上第一台永磁式直流发电机，这标志着电气时代的开始。

3）1873 年，比利时的古拉姆研制出世界上第一台环形电枢直流电动机，结束了蒸汽机时代。

直流电动机调速方便，控制灵活。但直流电动机由于本身结构上存在有机械换向器和电刷，所以给直流调速系统带来了以下主要缺点：

① 维修困难。

② 使用环境受限制，不适用于易燃、易爆及环境恶劣的地方。

③ 制造大容量、高转速及高电压的直流电动机比较困难。

4）1882 年，塞尔维亚（后加入美国籍）的特斯拉继爱迪生发明直流电（DC）后不久，即发明了交流电（AC），并制造出世界上第一台交流发电机，并于 1888 年发明了交流多相电力传输系统并因此获得美国专利。但交流电动机的调速性能较差，这促使人们研究交流系统的调速技术。

5）1920 年后即发现了变频调速的优越性。

6）20 世纪 60 年代电力电子技术得到快速发展，1957 年美国通用电气公司发明晶闸管，并于 1958 年投入商用，晶闸管的诞生为变频调速提供了可能。

**2. 变频器技术的发展阶段**

芬兰瓦萨控制系统有限公司（前身为瑞典的 STRONGB），于 20 世纪 60 年代成立，并于 1967 年开发出世界上第一台变频器，被称为变频器的鼻祖，开创了世界商用变频器的市场。之后变频器技术不断发展。按照变频器的控制方式，可划分为以下几个阶段：

（1）第一阶段：恒压频比 U/f 技术

U/f 控制就是保证输出电压跟频率成正比的控制，这样可以使电动机的磁通保持一定，避免弱磁和磁饱和现象的产生，多用于风机、泵类节能型变频器。日本于 20 世纪 80 年代，开发出电压空间矢量控制技术，后引入频率补偿控制。电压空间矢量的频率补偿方法不仅能

消除速度控制的误差，而且可以通过反馈估算磁链幅值，消除低速时定子电阻的影响，将输出电压、电流闭环，以提高动态的准确度和稳定度。

（2）第二阶段：矢量控制

20 世纪 70 年代，德国人（F. Blaschke）首先提出了矢量控制模型。矢量控制实现的基本原理是通过测量和控制异步电动机定子电流矢量，根据磁场定向原理分别对异步电动机的励磁电流和转矩电流进行控制，从而达到控制异步电动机转矩的目的。1992 年，西门子公司开发出了 6SE70 系列矢量控制的变频器，是矢量控制模型的代表产品。

矢量控制方式又有基于转差频率控制的矢量控制方式、无速度传感器矢量控制方式和有速度传感器的矢量控制方式等。这样就可以将一台三相异步电动机等效为直流电动机来控制，从而获得与直流调速系统同样的静、动态性能。矢量控制算法已被广泛地应用在 SIEMENS、ABB、GE、Fuji 和 SAJ 等国际化大公司的变频器上。

（3）第三阶段：直接转矩控制

直接转矩控制系统（Direct Torque Control 简称 DTC）是在 20 世纪 80 年代中期继矢量控制技术之后发展起来的一种高性能异步电动机变频调速系统。1977 年美国学者 A. B. Plunkett 在 IEEE 杂志上首先提出了直接转矩控制理论，1985 年，德国鲁尔大学的 Depenbrock 教授和日本的 Tankahashi 分别将“直接转矩控制”应用于实践，并取得成功。接着在 1987 年又把直接转矩控制推广到弱磁调速范围。不同于矢量控制，直接转矩控制具有鲁棒性强、转矩动态响应速度快及控制结构简单等优点，在很大程度上解决了矢量控制中结构复杂、计算量大以及对参数变化敏感等问题。直接转矩控制技术的主要问题是低速时转矩脉动大，其低速性能还是不能达到矢量控制的水平。

1995 年，美国 ABB 公司推出的 ACS600 直接转矩控制系列变频器是直接转矩控制的代表产品。

表 11-1 是 20 世纪 60 年代到 21 世纪初，变频器技术发展的历程。

**表 11-1　变频器技术发展的历程**

| 项　　目 | 60 年代 | 70 年代 | 80 年代 | 90 年代 | 21 世纪初 |
|---|---|---|---|---|---|
| 电动机控制算法 | $U/f$ 控制 | | 矢量控制 | 无速度矢量控制电流矢量 $U/f$ | 算法优化 |
| 功率半导体技术 | SCR（晶闸管） | GTR（电力晶体管） | IGBT（绝缘栅双极型晶体管） | IGBT 大容量 | 更大容量更高开关频率 |
| 计算机技术 | | | 单片机 DSP | 高速 DSP 专用芯片 | 更高速率和容量 |
| PWM 技术 | | PWM 技术 | SPWM 技术 | 空间电压矢量调制技术 | PWM 优化新一代开关技术 |
| 变频器的特点 | 大功率传动使用变频器，体积大，价格高 | 变频器体积缩小，开始在中小功率电动机上使用 | 超静音变频器开始流行，解决了 GTR 噪声问题。变频器性能大幅提升，大批量使用，取代直流调速 | | 未来发展方向是完美无谐波，如：矩阵式变频器 |

**3. 我国变频器技术发展现状**

20 世纪 80 年代，大量中小型日本品牌的变频器进入我国市场。90 年代后，欧美大容量变频器如西门子、ABB、AB、科比以及罗克韦尔等进入中国市场。

1989 年大连引入日本东芝第一条变频器生产线。

1991 年中国台湾普传进入内地，建设了许多合资公司、研发中心和合资厂，是当时最大最完善的营销网，各合资公司组装变频器机箱，主控制板在台湾制造，pi89 变频器是当时国内变频器市场上主流产品，但现在已被淘汰。

目前，国内有超过 200 多家厂家生产变频器，以森兰、英威腾和汇川为代表，技术水平较接近世界先进水平，但总市场份额只有 10% 左右。国产的变频器主要是交流 380V 的中小型变频器，且大部分产品为低压，而高压大功率的则很少，能够研制、生产、并提供服务的高压变频器厂商更少，不过是少数几个具备科研能力或资金实力强的企业。并且在技术方面，更是普遍采用 U/f 控制方式，对中、高压电动机进行的变频调速改造。我国高压变频器的品种和性能还处于发展的初步阶段，仍需大量从国外进口。这一现状主要表现在以下方面：

1）国外各大品牌的产品，加快了占领国内市场的步伐并将产品本地化，我国目前变频器市场较大，仍有巨大的发展潜力。

2）多数企业不具备足够资金进行科研和规模化生产，生产工艺相对落后，产品的技术含量低，品质有待提高，但总体上价格低廉。

3）国内高压变频器尚未形成一套完备的标准，产品差异性大，需要进一步完备、完善高压变频器的标准，同时，国产高压变频器的功率等级较低，一般不超过 3 500 kW。

4）高压变频器周边产业少，不够发达，约束了高压变频器的发展速度，很多变频器中的主要功率器件无法自主生产，如：驱动电路、电解电容等。

5）自主研发能力在逐步提高，与发达国家的技术差距在缩小，自主创新的技术和产品也逐步得到应用。

6）已经研制出具有瞬时掉电再恢复、故障再恢复等性能的变频器，同时正在研发能够进行四象限运行的高压变频器。

**4. 变频器技术存在的问题**

变频器在使用中存在的主要问题是干扰。电网是一个非常复杂的结构，电网谐波是对变频器产生干扰的主要干扰源。谐波源的产生主要是各种整流设备、交直流互换设备、电子电压调整设备、非线性负载以及照明设备等等。这些设备在起动和工作的时候，产生对电网的冲击波（即电磁干涉），使得电网中的电压、电流的波形发生一定的畸变，对电网中的其他设备产生的这种谐波影响，需要进行简单的处理，即在接入变频器处加装电源滤波器，滤去干扰波，使变频器尽可能不受电网中的这些谐波的影响，从而稳定工作。其次，另一种共模干涉，则通过变频器的控制线，对控制信号产生一定的干扰，影响其正常工作。

**5. 变频器的发展趋势**

随着节约环保型社会发展模式的提出，人们开始更多地关注生活环境的品质，因此，节能型、低噪声变频器是今后一段时间发展的总趋势。我国变频器的生产商家虽然不少，但是缺少统一的、具体的规范标准，使得产品差异性较大，且大部分采用了 U/f 控制和电压矢量控制，其准确度较低，动态性能不高，稳定性能较差，这些方面与国外同等产品相比有一定的差距。就变频器设备来说，其发展趋势主要表现在以下方面：

1）变频器将朝着高压大功率和低压小功率、小型化和轻型化的方向发展。

2）工业高压大功率变频器、民用低压中小功率变频器潜力巨大。

3）目前，IGBT、IGCT 以及 SGCT 仍将扮演主要的角色，SCR、GTO 将会退出变频器市场。

4）无速度传感器的矢量控制、磁通控制和直接转矩控制等技术的应用将趋于成熟。

5）全面实现数字化和自动化、参数自设定技术、过程自优化技术及故障自诊断技术。

6）高性能单片机的应用优化了变频器的性能，实现了变频器的高准确度和多功能。

7）相关配套行业正朝着专业化、规模化发展，社会分工逐渐明显。

8）伴随着节约型社会的发展，变频器在民用领域的使用会逐步得到推广。

### 11.1.2　变频器的分类

变频器发展到今天，生产厂家已经研制了多种适合不同用途的变频器，以下详细介绍变频器的分类。

**1. 按变换的环节分类**

1）交－直－交变频器。它先将工频交流电通过整流器变成直流电，然后再将直流电变换成频率、电压可调的交流电，又称间接式变频器，是目前广泛应用的通用型变频器。

2）交－交变频器。它将工频交流电直接变换成频率、电压可调的交流电，又称直接式变频器。主要用于大功率（500kW 以上）低速交流传动系统中，目前已经在轧机、鼓风机、破碎机、球磨机和卷扬机等设备中应用。这种变频器可用于异步电动机和同步电动机的调速控制。

这两种变频器的比较见表 11-2。

**表 11-2　交－直－交变频器和交－交变频器的比较**

| 交－直－交变频器 | 交－交变频器 |
| --- | --- |
| • 结构简单<br>• 输出频率变化范围大<br>• 功率因数高<br>• 谐波易于消除<br>• 可使用各种新型大功率器件 | • 过载能力强<br>• 效率高输出波形好<br>• 但输出频率低<br>• 使用功率器件多<br>• 输入无功功率大<br>• 高次谐波对电网影响大 |

**2. 按直流电源性质分类**

1）电压型变频器。电压型变频器特点是中间直流环节的储能元件采用大电容，负载的无功功率将由它来缓冲，直流电压比较平稳，直流电源内阻较小，相当于电压源，故称电压型变频器，常选用于负载电压变化较大的场合。这种变压器应用广泛。

2）电流型变频器。电流型变频器特点是中间直流环节采用大电感作为储能环节，缓冲无功功率，即扼制电流的变化，使电压接近正弦波，由于该直流内阻较大，故称电流源型变频器（电流型）。电流型变频器的特点（优点）是能扼制负载电流频繁而急剧的变化。常用于负载电流变化较大的场合。

**3. 按照用途分类**

可以分为通用变频器、高性能专用变频器、高频变频器、单相变频器和三相变频器等。此外，变频器还可以按输出电压调节方式分类、按控制方式分类、按主开关元器件分类以及按输入电压高低分类。

**4. 按变频器调压方法分类**

1）PAM 变频器。它是通过改变电压源 $U_d$或电流源 $I_d$ 的幅值进行输出控制的。这种变频器已很少使用了。

2）PWM 变频器。它是在变频器输出波形的每半个周期分割成许多脉冲，通过调节脉冲宽度和脉冲周期之间的“占空比”调节平均电压，其等值电压为正弦波，波形较平滑。

**5. 按控制方式分**

1）U/f 控制（VVVF 控制）变频器。U/f 控制就是保证输出电压跟频率成正比的控制。低端变频器都采用这种控制原理。

2）SF 控制（转差频率控制）变频器。转差频率控制是通过控制转差频率来控制转矩和电流，是高准确度的闭环控制，但通用性差，一般用于车辆控制。与 U/f 控制相比，其加减速特性和限制过电流的能力得到提高。另外，它有速度调节器，利用速度反馈构成闭环控制，速度的静态误差小。

3）VC（Vectory Control，矢量控制）变频器。实现矢量控制的基本原理是通过测量和控制异步电动机定子电流矢量，根据磁场定向原理分别对异步电动机的励磁电流和转矩电流进行控制，从而达到控制异步电动机转矩的目的。一般用在高准确度要求的场合。

4）直接转矩控制。简单地说，就是将交流电动机等效为直流电动机进行控制。

**6. 按国际区域分类**

1）中国内地变频器品牌。如安邦信、汇川、浙江三科、欧瑞传动、森兰、英威腾、蓝海华腾、迈凯诺、伟创以及易泰帝等，国产的品牌已经超过 200 家。另外中国港台变频器品牌有：台达、普传、台安、东元、美高。

2）欧美变频器品牌。如西门子、科比、伦茨、施耐德、ABB、丹佛斯、罗克韦尔、伟肯、AB 及西威。

3）日本变频器品牌。如富士、三菱、安川、三垦、日立、欧姆龙、松下电器、松下电工、东芝以及明电舍。

4）韩国变频器品牌。如 LG 、现代、三星和收获。

**7. 按电压等级分类**

1）高压变频器：3 kV、6 kV 及 10 kV 。

2）中压变频器：660 V、1140 V 。

3）低压变频器：220 V、380 V。

**8. 按电压性质分类**

1）交流变频器：AC－DC－AC（交－直－交）、AC－AC（交－交）。

2）直流变频器：DC－AC（直－交）。

## 11.2 西门子 MM440 变频器使用简介

### 11.2.1 认识变频器

**1. 初识变频器**

变频器一般是利用电力半导体器件的通断作用将工频电源变换为另一频率的电能控制装

置。变频器有着“现代工业维生素”之称，在节能方面的效果不容忽视。随着各界对变频器节能技术和应用等方面认识的逐渐加深，我国变频器市场变得异常活跃。

变频器产生的最初目的是速度控制，应用于印刷、电梯、纺织、机床和生产流水线等行业。而目前相当多的运用是以节能为目的。由于中国是能源消耗大国，而中国的能源储备又相对贫乏，因此国家大力提倡各种节能措施，其中着重推荐了变频器调速技术。在水泵和中央空调等领域，变频器可以取代传统的限流阀和回流旁路技术，充分发挥节能效果；在火电、冶金、矿山及建材行业，高压变频调速的交流电动机系统的经济价值正在得以体现。

变频器是一种高技术含量、高附加值且高效益回报的高科技产品，符合国家产业发展政策。在过去的二十几年间，我国变频器行业从起步阶段到目前的趋于成熟，发展十分迅速。进入 21 世纪以来，我国中、低压变频器市场的增长速度超过了 20%，远远高于近几年的 GDP 增长水平。

从产品优势角度看，通过高质量地控制电动机转速，提高制造工艺水准，变频器不但有助于提高制造工艺水平（尤其在精细加工领域），而且可以有效地节约电能，是目前最理想、最有前途的电动机节能设备。

从变频器行业所处的宏观环境看，无论是国家中长期规划、短期的重点工程、政策法规和国民经济整体运行趋势，还是人们节能环保意识的增强、技术的创新及发展高科技产业的要求，从国家相关部委到各相关行业，变频器都受到了广泛的关注，市场吸引力巨大。西门子变频器外形如图 11-1 所示。



图 11-1　变频器外形

**2. 交－直－交变频调速的原理**

交－直－交变频调速的原理如图 11-2 所示，交－直－交变频调速就是变频器先将工频交流电整流成直流电，逆变器在微控制器（如 DSP）的控制下，将直流电逆变成不同频率的交流电。目前市面上的变频器多是这种工作原理。

图 11-2 中的 R0 起限流作用，当 R、S 和 T 端子上的电源接通时，R0 接入电路，以限制起动电流。延时一段时间后，晶闸管 VT 导通，将 R0 短路，避免造成附加损耗。$R_t$ 为能耗制动电阻，当制动时，异步电动机进入发动机状态，逆变器向电容 C 反向充电，当直流回路的电压，即电阻 R1、R2 上的电压升高到一定的值时（图中实际上测量的是电阻 R2 的电压），通过泵升电路使开关器件 Vb 导通，这样电容 C 上的电能就消耗在制动电阻 Rt 上。通常为了散热，制动电阻 Rt 安装在变频器外侧。电容 C 除了参与制动外，在电动机运行时，主要起滤波作用。顺便指出起滤波作用是电容器的变频器称为电压型变频器；起滤波作用是电感器的变频器称为电流型变频器。比较多见的是电压型变频器。

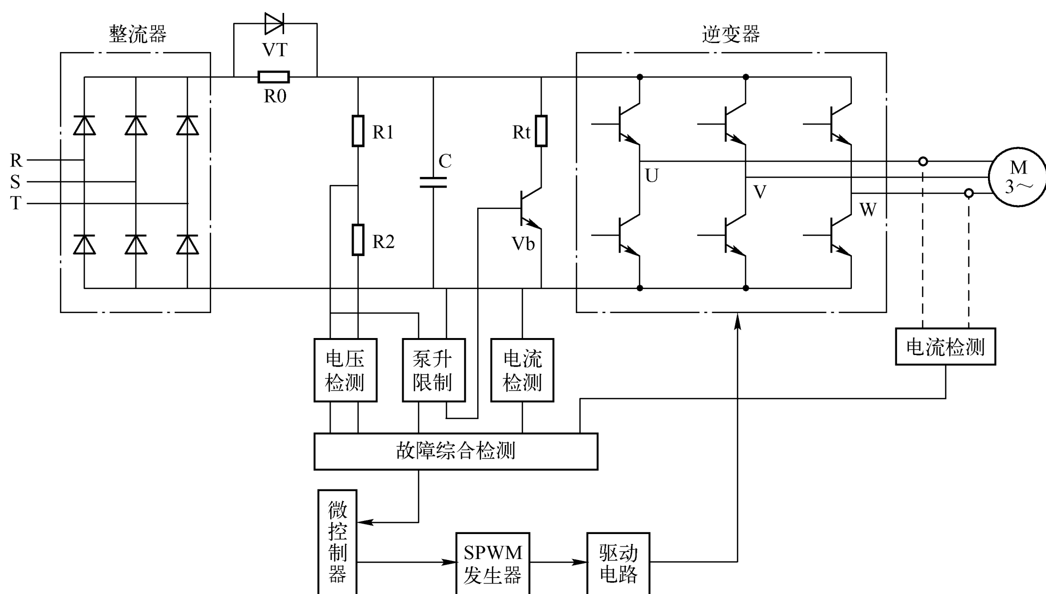



图 11-2　变频调速的原理图

微控制器经运算输出控制正弦信号后，经过 SPWM（正弦脉宽调制）发生器调制，再由驱动电路放大信号，放大后的信号驱动 6 个功率晶体管，产生三相交流电压 U、V、W 驱动电动机运转。

**【例 11-1】**如图 11-2 所示，若将变频器的动力线的输入和输出接反是否可行？若不可行有什么后果？

**解：**将变频器的动力线的输入和输出接反是不允许的，可能发生爆炸。

### 11.2.2　西门子低压变频器简介

西门子公司生产的变频器品种较多，以下简单介绍西门子低压变频器的产品系列。

**1. MM4 系列变频器**

MM4 系列变频器分为 4 个子系列，分别如下：

1）MM410：解决简单驱动问题的解决方案，功率范围小。

2）MM420：I/O 点数少，不支持矢量控制，无自由功能块使用，功率范围小。

3）MM430：风机水泵专用，不支持矢量控制。

4）MM440：支持矢量控制，有制动单元，有自由功能块使用，功能相对强大。

MM4 系列变频器有一定的市场占有率，但有被 SINAMICS G120 系列取代的趋势。

**2. SIMOVERT Master Drives，6SE70 工程型变频器**

其控制面板采用 CUVC，可实现变频调速、力矩控制和四象限工作，但有被 SINAMICS S120 系列取代的趋势。

**3. SINAMICS 系列变频器**

SINAMICS 系列变频器子系列较多，功能也较强大，具体简介如下：

1）SINAMICS G120：MM4 系列的升级版，含 CU（控制单元）和 PM（功率模块）两部分，可四象限工作，功能强大。

2）SINAMICS S120：6SE70 系列变频器的升级版，控制面板是 CU320，功能强大。可以驱动交流异步电动机、交流同步电动机和交流伺服电动机。

3）SINAMICS G120D：提高了 SINAMICS G120 系列变频器的防护等级，可以达到 IP65，但功率范围有限。

4）SINAMICS V50：MM4 系列变频器的柜机。

5）SINAMICS G150：V50 系列变频器的升级版，功率范围大。

6）SINAMICS V60 和 V80：是针对步进电动机而推出的两款产品，当然也可以驱动伺服电动机。只能接收脉冲信号。有人称其为简易的伺服驱动器。

7）SINAMICS V90：是针对步进电动机而推出的产品，当然也可以驱动伺服电动机。能接收脉冲信号，也支持 USS 和 MODBUS 总线。运动控制时配合西门子的 S7－200 SMART 使用，很方便。

### 11.2.3　西门子 MM440 变频器使用简介

**1. 初识西门子 MM440 变频器**

西门子 MM440 变频器由微处理器控制，并采用具有现代先进技术水平的绝缘栅双极型晶体管（IGBT）作为功率输出器件，它具有很高的运行可靠性和功能多样性。脉冲宽度调制的开关频率也是可选的，降低了电动机运行的噪声。

MM440 变频器的框图如图 11-3 所示，控制端子定义见表 11-3。

**表 11-3　MM440 控制端子表**

| 端子序号 | 端子名称 | 功　能 | 端子序号 | 端子名称 | 功　能 |
|---|---|---|---|---|---|
| 1 | － | 输出 +10 V | 16 | DIN5 | 数字输入 5 |
| 2 | － | 输出 0 V | 17 | DIN6 | 数字输入 6 |
| 3 | ADC1 + | 模拟输入 1（+） | 18 | DOUT1/NC | 数字输出 1/常闭触点 |
| 4 | ADC1 － | 模拟输入 1（－） | 19 | DOUT1/NO | 数字输出 1/常开触点 |
| 5 | DIN1 | 数字输入 1 | 20 | DOUT1/COM | 数字输出 1/转换触点 |
| 6 | DIN2 | 数字输入 2 | 21 | DOUT2/NO | 数字输出 2/常开触点 |
| 7 | DIN3 | 数字输入 3 | 22 | DOUT2/COM | 数字输出 2/转换触点 |
| 8 | DIN4 | 数字输入 4 | 23 | DOUT3NC | 数字输出 3 常闭触点 |
| 9 | － | 隔离输出 +24 V/max. 100 mA | 24 | DOUT3NO | 数字输出 3 常开触点 |
| 10 | ADC2 + | 模拟输入 2（+） | 25 | DOUT3COM | 数字输出 3 转换触点 |
| 11 | ADC2 － | 模拟输入 2（－） | 26 | DAC2 + | 模拟输出 2（+） |
| 12 | DAC1 + | 模拟输出 1（+） | 27 | DAC2 － | 模拟输出 2（－） |
| 13 | DAC1 － | 模拟输出 1（－） | 28 | － | 隔离输出 0 V/max. 100 mA |
| 14 | PTCA | 连接 PTC/KTY84 | 29 | P + | RS 485 |
| 15 | PTCB | 连接 PTC/KTY84 | 30 | P － | RS 485 |

MM440 变频器的核心部件是 CPU 单元。根据设定的参数，经过运算，输出控制正弦波信号，正弦波信号再经过 SPWM 调制、放大，输出正弦交流电，驱动三相异步电动机运转。

MM440 变频器是一个智能化的数字变频器，在基本操作板上可进行参数设置，参数可分为 4 个级别：

图 11-3　MM440 变频器的框图

① 标准级，可以访问经常使用的参数。

② 扩展级，允许扩展访问参数范围，例如变频器的 I/O 功能。

③ 专家级，只供专家使用，即高级用户。

④ 维修级，只供授权的维修人员使用，具有密码保护。

【关键点】对于一般的用户，将变频器设置成标准级或者扩展级即可。

基本操作面板（BOP）的外形如图 11-4 所示，利用基本操作面板可以设置变频器的参数。BOP 具有 7 段显示的 5 位数字，可以显示参数的序号和数值、报警和故障信息，以及设定值和实际值。参数的信息不能用 BOP 存储。BOP 上的按钮的功能见表 11-4。




图 11-4　基本操作面板（BOP）的外形

**表 11-4　BOP 上的按钮的功能**

| 显示/按钮 | 功　　能 | 功能的说明 |
| --- | --- | --- |
| r0000 | 状态显示 | LED 显示变频器当前的设定值 |
| I | 起动变频器 | 按此按钮起动变频器。默认值运行时此按钮是被封锁的。为了使此按钮有效，应设定 P0700 = 1 |
| 0 | 停止变频器 | OFF1：按此按钮，变频器将按选定的斜坡下降速率减速停车。默认值运行时此按钮被封锁，为了允许此按钮操作，应设定 P0700 = 1。OFF2：按此按钮两次（或一次，但时间较长），电动机将在惯性作用下自由停车，此功能总是“使能”的 |
| (换向按钮) | 改变电动机的旋转方向 | 按此按钮可以改变电动机的旋转方向。电动机的反向用负号（ - ）表示或用闪烁的小数点表示。在默认设定时此按钮被封锁。为使此按钮有效，应先按“起动电动机”按钮 |
| jog | 电动机点动 | 在“准备合闸”状态下按压此按钮，则电动机起动并运行在预先设定的点动频率。当释放此按钮，电动机停车。当电动机正在旋转时，此按钮无功能 |
| Fn | 功能 | 此按钮用于浏览辅助信息<br>变频器运行过程中，在显示任何一个参数时按下此按钮并保持不动 2 s，将显示以下参数值（在变频器运行中，从任何一个参数开始）：<br>1. 直流回路电压（用 d 表示，单位：V）。<br>2. 输出电流（A）。<br>3. 输出频率（Hz）。<br>4. 输出电压（用 o 表示，单位为 V）。<br>5. 由 P0005 选定的数值（如果 P0005 选择显示上述参数中的任何一个（3、4 或 5），这里将不再显示）。<br>连续多次按下此按钮，将轮流显示以上参数<br>跳转功能，在显示任何一个参数（rXXXX 或 PXXXX）时，短时间按下此按钮，将立即跳转到 r0000，如果需要的话，可以接着修改其他的参数。跳转到 r0000 后，按此按钮将返回原来的显示点 |
| P | 访问参数 | 按此按钮即可访问参数 |
| ▲ | 增加数值 | 按此按钮即可增加面板上显示的参数数值 |
| ▼ | 减少数值 | 按此按钮即可减少面板上显示的参数数值 |
| Fn + P | AOP 菜单 | 调出 AOP（高级操作面板）菜单提示（仅用于 AOP） |

### 2. MM440 变频器 BOP 频率给定

以下通过例子介绍 MM440 变频器 BOP 频率给定的过程。

【例 11-2】一台 MM440 变频器配一台西门子三相异步电动机，已知电动机的技术参数，功率为 0.75 kW，额定转速为 1400 r/min，额定电压为 380 V，额定电流为 2.05 A，额定频率为 50 Hz，试用 BOP 设定电动机的运行频率为 10 Hz。

**解：**

1）先介绍如何设定参数，以下通过将参数 P1000 的第 0 组参数，即设置 P1000[0]=1 的过程为例，讲解一个参数的设置方法。参数的设定方法见表 11-5。

**表 11-5　参数的设定方法**

| 序号 | 操作步骤 | BOP 显示 |
|---|---|---|
| 1 | 按Ⓟ按钮，访问参数 | r0000 |
| 2 | 按▲按钮，直到显示 P1000 | P1000 |
| 3 | 按Ⓟ按钮，显示 in000，即 P1000 的第 0 组值 | in000 |
| 4 | 按Ⓟ按钮，显示当前值 2 | 2 |
| 5 | 按▼按钮，达到所要求的数值 1 | 1 |
| 6 | 按Ⓟ按钮，存储当前设置 | P1000 |
| 7 | 按Fn按钮，显示 r0000 | r0000 |
| 8 | 按Ⓟ按钮，显示频率 | 10.00 |

2）完整的设置过程。按照表 11-6 中的步骤进行。

**表 11-6　设置过程**

| 步骤 | 参数及设定值 | 说　明 | 步骤 | 参数及设定值 | 说　明 |
|---|---|---|---|---|---|
| 1 | P0003=2 | 扩展级 | 8 | P1000=1 | 频率源为 BOP |
| 2 | P0010=1 | 为 1 才能修改电动机参数 | 9 | P0700=1 | 命令源（起停）为 BOP |
| 3 | P0304=380 | 额定电压 | 10 | P1080=0 | 最小频率 |
| 4 | P0305=2.05 | 额定电流 | 11 | P1082=50 | 最大频率 |
| 5 | P0307=0.75 | 额定功率 | 12 | P1120=10 | 从静止到达最大频率所需时间 |
| 6 | P0311=1400 | 额定转速 | 13 | P1121=10 | 从最大频率到停止所需时间 |
| 7 | P0010=0 | 运行和设定变频器参数时，必须为 0 | | | |

3）起停控制。按下基本操作面板上的按钮，三相异步电动机起动，稳定运行的频率为 10Hz；当按按钮时，电动机停机。

【关键点】初学者在设置参数时，有时进行了错误的设置，但又不知道在什么参数的设置上出错，通常这种情况下可以对变频器进行复位，常规的变频器都有这个功能，复位后变频器的所有参数变成出厂时的设定值，但工程中正在使用的变频器要谨慎使用此功能。西门子 MM440 的复位方法是：先将 P0010 设置为 30，再将 P0970 设置为 1，变频器上的显示器中闪烁的“busy”消失后，变频器成功复位。

## 11.3　变频器多段频率给定

变频器频率给定一般有 4 种方法，操作面板频率给定、模拟量频率给定、多段频率给定

和通信频率给定，将在本节和后续章节中分别介绍。

用基本操作面板进行手动频率给定方法简单，对资源消耗少，但这种频率给定方法对于操作者来说比较麻烦，而且不容易实现自动控制，而 PLC 控制的多段频率给定和通信频率给定，就容易实现自动控制，以下将通过两个例题来介绍 MM440 变频器的多段频率给定。

**【例 11-3】** 有一台 MM440 变频器，接线图如图 11-5 所示，当按钮 SA1 接通时，三相异步电动机以 5 Hz 正转，当按钮 SA2 接通时，三相异步电动机以 10 Hz 正转，当按钮 SA3 接通时，三相异步电动机以 15 Hz 反转，已知电动机的技术参数，功率为 0.75 kW，额定转速为 1400 r/min，额定电压为 380 V，额定电流为 2.05 A，额定频率为 50 Hz，请设计方案。

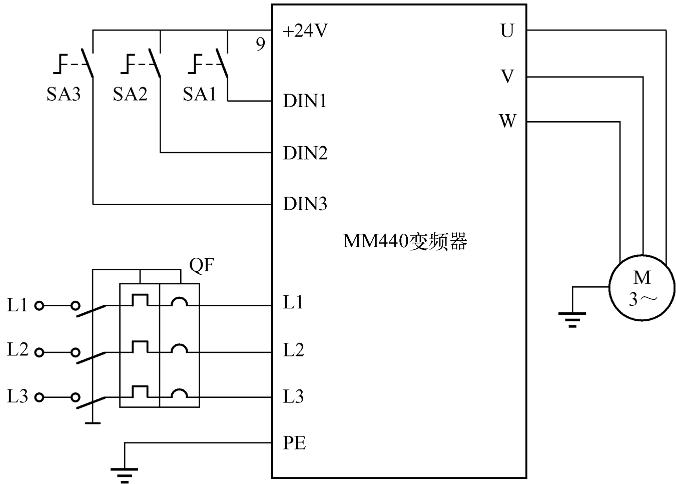



图 11-5 接线图

**解：** 多段频率给定时，当接通按钮 SA1 时，DIN1 端子与变频器的 +24 V（端子 9）连接时对应一个频率，频率值设定在 P1001 中；当接通按钮 SA2 时，DIN2 端子与变频器的 +24 V（端子 9）连接时再对应一个频率，频率值设定在 P1002 中；当接通按钮 SA3 时，DIN3 端子与变频器的 +24 V 接通，对应一个频率，频率值设定在 P1003 中。变频器参数见表 11-7。

**表 11-7 变频器参数**

| 序号 | 变频器参数 | 出 厂 值 | 设 定 值 | 功 能 说 明 |
|---|---|---|---|---|
| 1 | P0304 | 380 | 380 | 电动机的额定电压（380 V） |
| 2 | P0305 | 2.05 | 2.05 | 电动机的额定电流（2.05 A） |
| 3 | P0307 | 0.75 | 0.75 | 电动机的额定功率（0.75 W） |
| 4 | P0310 | 50.00 | 50.00 | 电动机的额定频率（50 Hz） |
| 5 | P0311 | 1400 | 1400 | 电动机的额定转速（1400 r/min） |
| 6 | P1000 | 2 | 3 | 频率源（固定频率设定） |
| 7 | P1080 | 0 | 0 | 电动机的最小频率（0 Hz） |
| 8 | P1082 | 50 | 50.00 | 电动机的最大频率（50 Hz） |
| 9 | P1120 | 10 | 10 | 斜坡上升时间（10 s） |
| 10 | P1121 | 10 | 10 | 斜坡下降时间（10 s） |
| 11 | P0700 | 2 | 2 | 选择命令源（由端子排输入） |

（续）

| 序号 | 变频器参数 | 出　厂　值 | 设　定　值 | 功 能 说 明 |
|---|---|---|---|---|
| 12 | P0701 | 1 | 16 | 固定频率设定值（直接选择 + ON） |
| 13 | P0702 | 12 | 16 | 固定频率设定值（直接选择 + ON） |
| 14 | P0703 | 9 | 2 | 反转 |
| 15 | P1001 | 0. 00 | 5 | 固定频率 1（5 Hz） |

**【例 11-4】**有一台小车，由 CPU314C－2DP 控制一台 MM440 变频器拖动，原始位置有限位开关 SQ1，原理图如图 11-6 所示。已知电动机的技术参数，功率为 0. 75 kW，额定转速为 1400 r/min，额定电压为 380 V，额定电流为 2. 05 A，额定频率为 50 Hz。系统有两种工作模式。

1）自动模式时，当在原始位置压下起动按钮 SB1 时，三相异步电动机以 30 Hz 正转，驱动小车前进，碰到限位开关 SQ2 后，三相异步电动机以 40 Hz 反转，小车后退，当碰到减速限位开关 SQ3 后时，三相异步电动机以 10 Hz 反转，小车减速后退，碰到原始位置有限位开关 SQ1，小车停止。压下停止 SB2 按钮时，小车完成一个工作循环后，停机。

2）手动模式时，有前进和后退点动按钮，点动的频率都是 30 Hz。

3）任何时候，压下急停 SB3 按钮，系统立即停机，请设计方案，并编写程序。

**解：**

**1. 主要软硬件配置**

① 1 套 STEP V5. 5 SP4。

② 1 台 MM440 变频器。

③ 1 台 CPU314C－2DP。

④ 1 台电动机。

⑤ 1 根编程电缆。

硬件配置如图 11-6 所示。

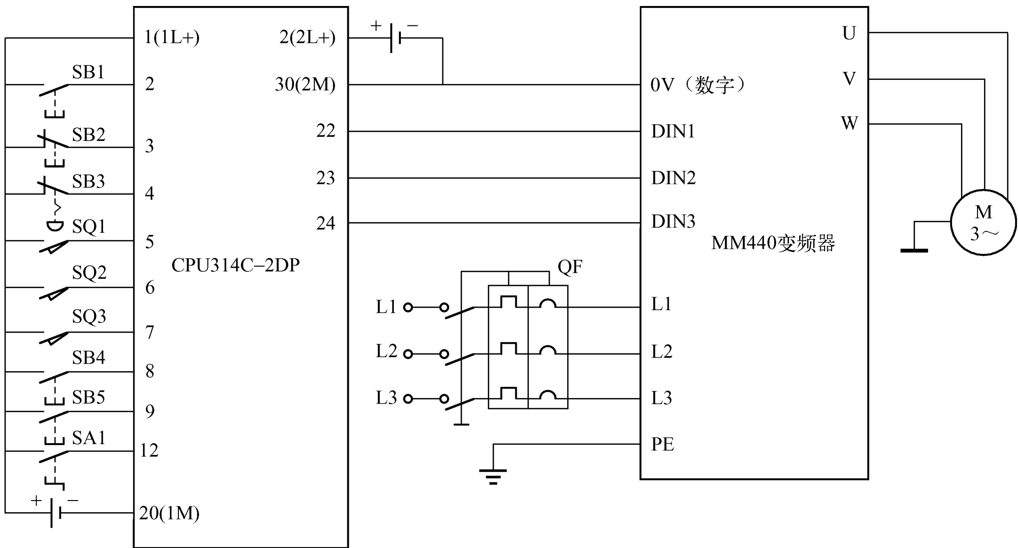



图 11-6　接线图

**2. 参数的设置**

多段频率给定时，当 DIN1 端子高电平时对应一个频率，当 DIN2 端子高电平时对应一个频率，当 DIN1 和 DIN2 端子同时高电平时再对应一个频率，DIN3 端子高电平时为反转，DIN3 端子低电平时为正转。变频器参数见表 11–8。

**表 11–8　变频器参数**

| 序号 | 变频器参数 | 出厂值 | 设定值 | 功能说明 |
|---|---|---|---|---|
| 1 | P0304 | 380 | 380 | 电动机的额定电压（380 V） |
| 2 | P0305 | 2. 05 | 2. 05 | 电动机的额定电流（2. 05 A） |
| 3 | P0307 | 0. 75 | 0. 75 | 电动机的额定功率（750 W） |
| 4 | P0310 | 50. 00 | 50. 00 | 电动机的额定频率（50 Hz） |
| 5 | P0311 | 1400 | 1400 | 电动机的额定转速（1400 r/min） |
| 6 | P1000 | 2 | 3 | 固定频率设定 |
| 7 | P1080 | 0 | 0 | 电动机的最小频率（0 Hz） |
| 8 | P1082 | 50 | 50. 00 | 电动机的最大频率（50 Hz） |
| 9 | P1120 | 10 | 10 | 斜坡上升时间（10 s） |
| 10 | P1121 | 10 | 10 | 斜坡下降时间（10 s） |
| 11 | P0700 | 2 | 2 | 选择命令源（由端子排输入） |
| 12 | P0701 | 1 | 16 | 固定频率设定值（直接选择 + ON） |
| 13 | P0702 | 12 | 16 | 固定频率设定值（直接选择 + ON） |
| 14 | P0703 | 9 | 12 | 反转 |
| 15 | P1001 | 0. 00 | 30 | 固定频率 1（30 Hz） |
| 16 | P1002 | 5. 00 | 10 | 固定频率 2（10 Hz） |

修改参数 P0701，对应设定数字输入 1（DIN1）的功能；修改参数 P0702，对应设定数字输入 2（DIN2）的功能，依次类推。

**3. 编写程序**

梯形图程序如图 11–7 所示。

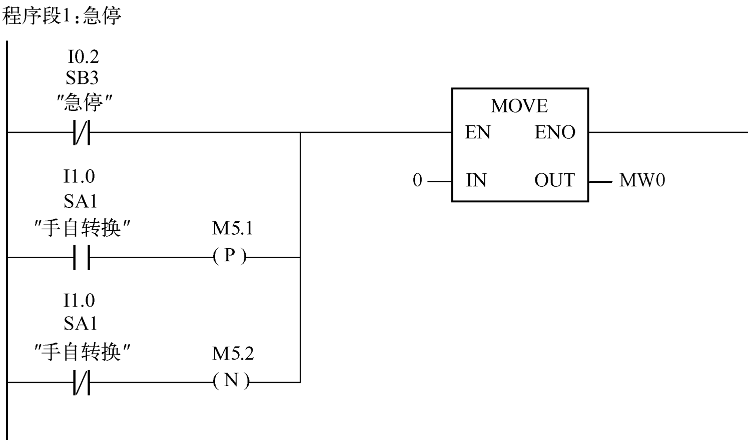
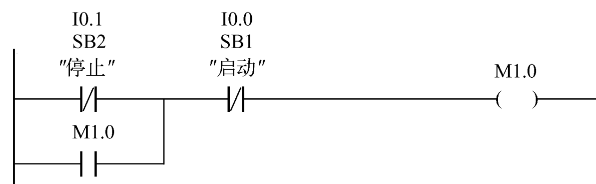
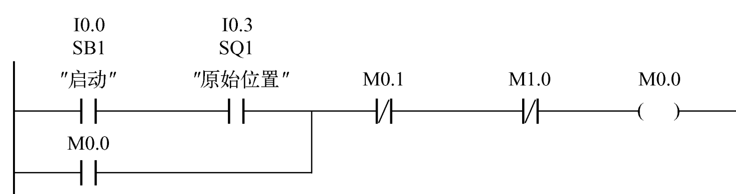



图 11–7　梯形图

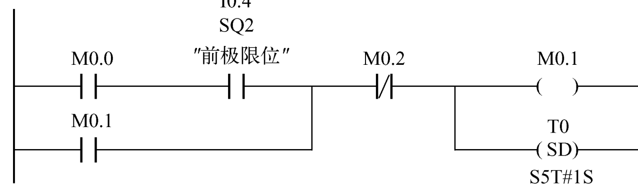
程序段2:停止



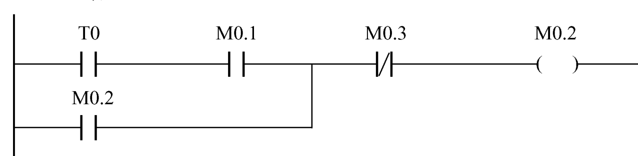

程序段3:标题



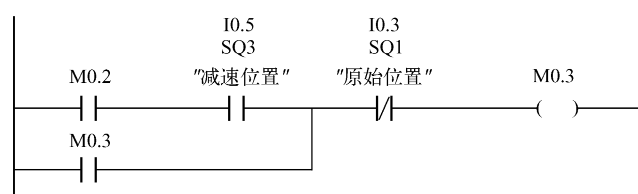

程序段4:标题




程序段5:标题




程序段6:标题




程序段7：标题：

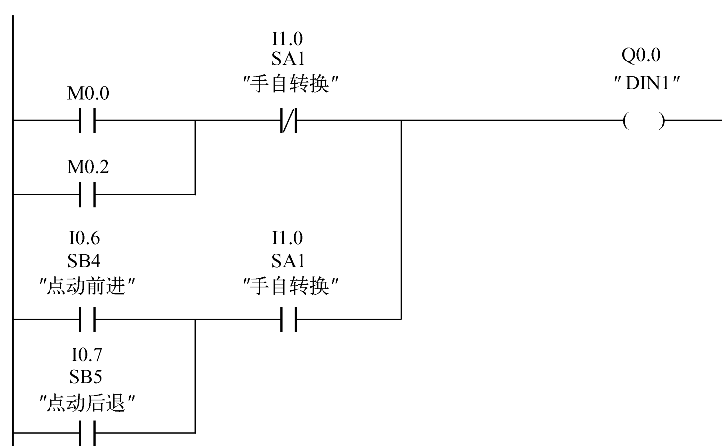



图 11-7　梯形图（续）

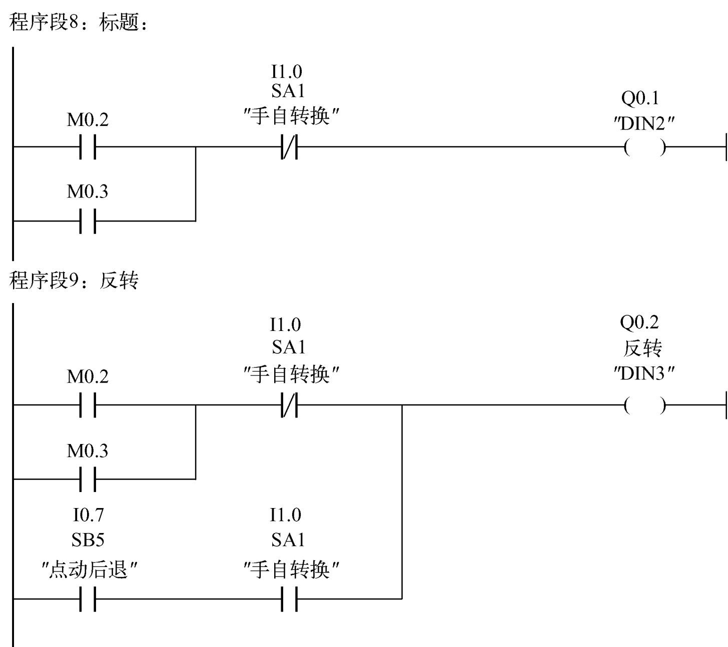



图 11-7　梯形图（续）

## 11.4　变频器的模拟量频率给定

虽然 BOP 模式的频率给定简单易行，但每次改变频率需要手动设置，不易实现自动控制，而模拟量频率给定可以比较方便地实现自动控制和无级调速，因而在工程中比较常用，但模拟量频率给定一般要用到模拟量模块，相对而言，控制成本稍高。

由于使用到 S7－300/400，所以绝大部分时候需要用到通信，因此通信频率给定较为常用，但是一旦通信系统中断，变频器也随之停止运行，而在某些场合，如通风场合，风机长时间停机是不允许的，这是通信频率给定的弱点，而模拟量频率给定却没有这个缺点，当通信系统中断后，只要把变频器用“手/自转换开关”切换到“手动”档，变频器仍可照常工作。以下用两个例子介绍模拟量频率给定。

**【例 11-5】**要对一台变频器进行电压信号模拟量频率给定，已知电动机的技术参数，功率为 0.75 kW，额定转速为 1400 r/min，额定电压为 380 V，额定电流为 2.05 A，额定频率为 50 Hz。设计电气控制系统，并设定参数。

**解：**电气控制系统如图 11-8 所示，只要调节电位器就可以对电动机进行无级调速，参数设定见表 11-9。

**表 11-9　变频器参数表**

| 序号 | 变频器参数 | 出 厂 值 | 设 定 值 | 功 能 说 明 |
|---|---|---|---|---|
| 1 | P0304 | 380 | 380 | 电动机的额定电压（380 V） |
| 2 | P0305 | 2.05 | 2.05 | 电动机的额定电流（2.05 A） |
| 3 | P0307 | 0.75 | 0.75 | 电动机的额定功率（0.75 kW） |

（续）

| 序号 | 变频器参数 | 出　厂　值 | 设　定　值 | 功 能 说 明 |
|---|---|---|---|---|
| 4 | P0310 | 50. 00 | 50. 00 | 电动机的额定频率（50 Hz） |
| 5 | P0311 | 0 | 1400 | 电动机的额定转速（1400 r/min） |
| 6 | P0700 | 2 | 2 | 选择命令源（由端子排输入） |
| 7 | P0756 | 0 | 0 | 选择 ADC 的类型（电压信号） |
| 8 | P1000 | 2 | 2 | 频率源（模拟量） |
| 9 | P701 | 1 | 1 | 数字量输入 1 |

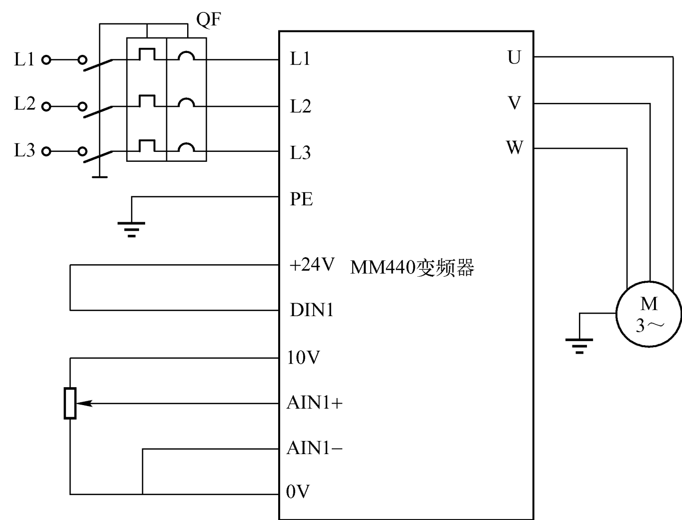



图 11-8　电气原理图

**【例 11-6】** 用一台触摸屏、PLC 对变频器进行频率给定（如触摸屏中输入 1400 r/min，则电动机以 50 Hz 旋转），已知电动机的技术参数，功率为 0. 75 kW，额定转速为 1400 rpm，额定电压为 380 V，额定电流为 2. 05 A，额定频率为 50 Hz。请设计电气控制系统，并编写程序。

**解：**

**1. 软硬件配置**

① 1 套 STEP 7 V5. 5 SP4。

② 1 台 MM440 变频器。

③ 1 台 CPU 314C－2DP。

④ 1 台电动机。

⑤ 1 根编程电缆。

⑥ 1 台 SM334。

⑦ 1 台 HMI。

原理图如图 11-9 所示。

**2. 硬件组态**

硬件组态如图 11－10 所示，可以看到 SM334 的模拟量输出的第 0 通道的地址为 PQW256。双击“AI4/AO02”，选择“输出”选项卡，选择通道 0 的输出范围，并取消通道 1，如图 11-11 所示。

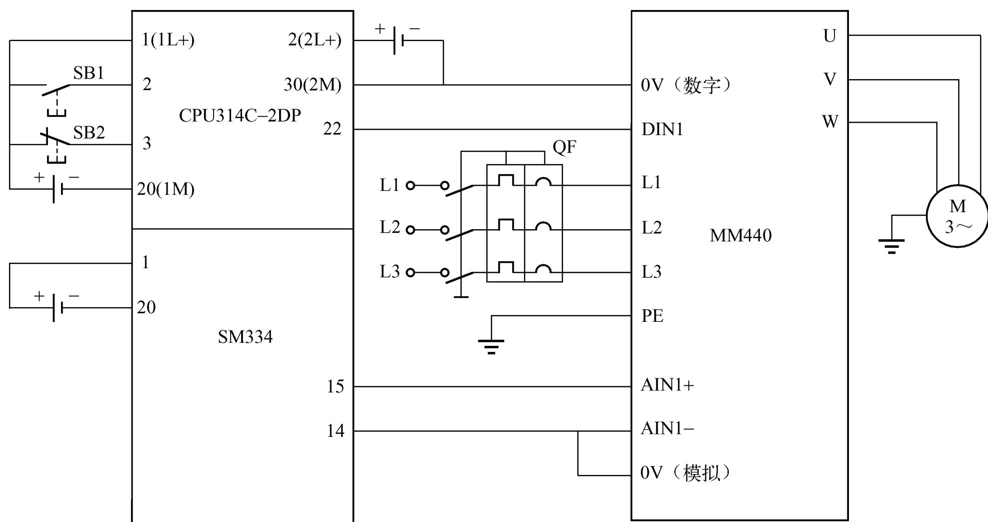



图 11-9　模拟量频率给定

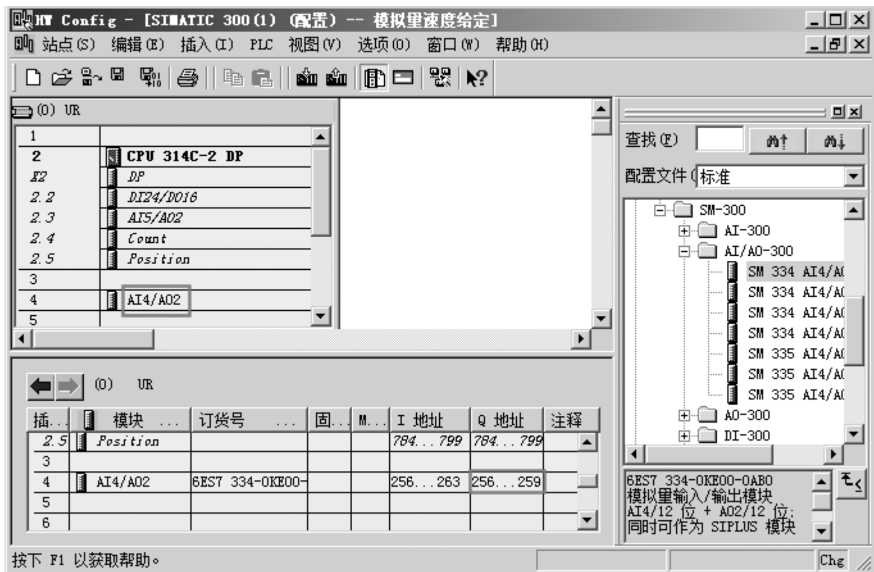



图 11-10　硬件组态（1）

### 3. 设定变频器的参数

先查询 MM440 变频器的说明书，再依次在变频器中设定表 11-10 中的参数。

**表 11-10　变频器参数表**

| 序号 | 变频器参数 | 出 厂 值 | 设 定 值 | 功 能 说 明 |
|---|---|---|---|---|
| 1 | P0304 | 380 | 380 | 电动机的额定电压（380 V） |
| 2 | P0305 | 2. 05 | 2. 05 | 电动机的额定电流（2. 05 A） |
| 3 | P0307 | 0. 75 | 0. 75 | 电动机的额定功率（0. 75 kW） |
| 4 | P0310 | 50. 00 | 50. 00 | 电动机的额定频率（50 Hz） |
| 5 | P0311 | 1400 | 1400 | 电动机的额定转速（1400 r/min） |

（续）

| 序号 | 变频器参数 | 出厂值 | 设定值 | 功能说明 |
|---|---|---|---|---|
| 6 | P0700 | 2 | 2 | 选择命令源（由端子排输入） |
| 7 | P0756 | 0 | 0 | 选择 ADC 的类型（电压信号） |
| 8 | P1000 | 2 | 2 | 频率源（模拟量） |
| 9 | P701 | 1 | 1 | 数字量输入 1 |




图 11-11　硬件组态（2）

**【关键点】** P0756 设定成 0，表示电压信号对变频器频率给定，默认是电压信号。此外，还要将 I/O 控制板上的 DIP 开关设定为“OFF”。

**4. 编写程序，并将程序下载到 PLC 中**

在组织块 OB1 输入如图 11-12 程序所示的梯形图，其功能是发送起停信号。在组织块 OB35 输入如图 11-13 程序所示的梯形图，其功能是每 100 ms 向变频器发送一个模拟量，其中 MW20 中的数据由触摸屏提供。

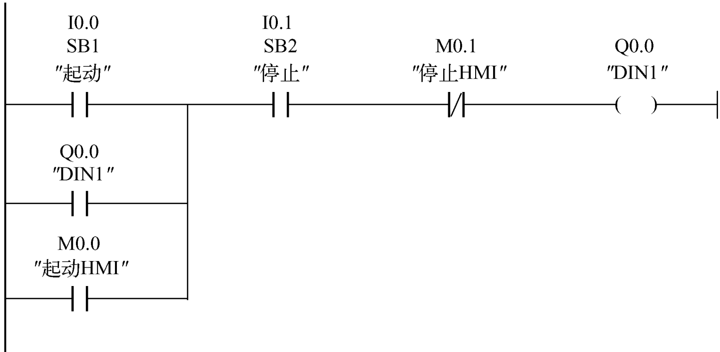
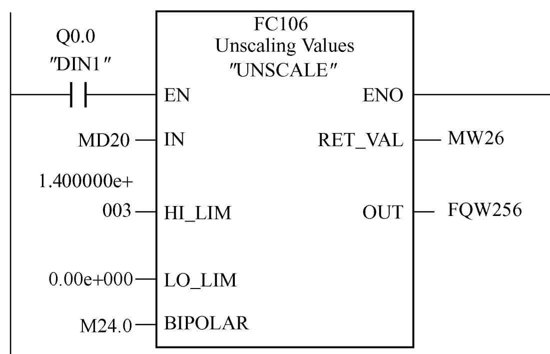



图 11-12　程序（组织块 OB1）

程序段1:标题:




程序段2:标题:

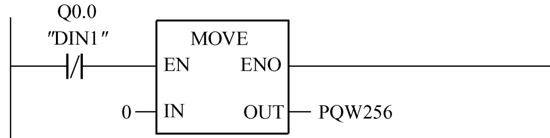



图 11-13　OB35 中的程序

# 11.5　变频器的通信频率给定

通信频率给定既可实现无级调速，也可实现自动控制，应用灵活方便。S7－300/400 与 MM440 变频器可进行 USS 和 PROFIBUS 通信。MM440 变频器也只能作 PROFIBUS－DP 从站，不能作 PROFIBUS－DP 主站，而且 MM440 变频器用作从站时要配置通信模板。

## 11.5.1　MM440 变频器通信的基本知识

MM440 变频器既支持和主站的周期性数据通信，也支持和主站的非周期性数据通信，也就是说 S7－300 可以使用功能块 SFC14/SFC15 读取和修改 MM440 的参数值，调用一次可以读取或者修改一个参数。同时也可以使用功能块 SFC58/SFC59 或者 SFB52/SFB53 读取或者修改 MM440 的参数，一次最多可以读取或者修改 39 个参数。

有效的数据块分成两个区域即 PKW 区（参数识别 ID－数值区）和 PZD 区（过程数据)，有效数据字符如图 11-14 所示。

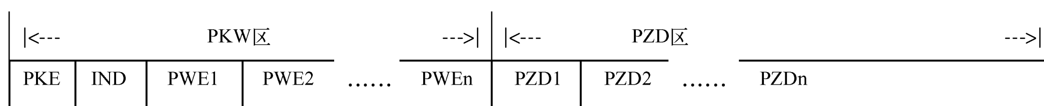



图 11-14　有效数据字符

PKW 区说明（参数识别 ID－数值区）PKW 接口的处理方式。PKW 接口并非物理意义上的接口，而是一种机理。这一机理确定了参数在两个通信伙伴之间（例如控制装置与变频器）的传输方式，例如参数数值的读和写。

PKW 区前两个字（即 PKE 和 IND）的信息是关于主站请求的任务（任务识别标记 ID）或应答报文的类型（应答识别标记 ID）。PKW 区的第三、第四个字规定报文中要访问的变

频器的参数号（PNU）。PNU 的编号与 MICROMASTER4 的参数号相对应。例如 1082 = P1082 = Fmax。第一个字 PKE 见表 11-11，第二个字 IND 见表 11-12。

**表 11-11　第一个字 PKE**

| 第 1 个字 16 位 = PKE = 参数识别标记 ID | | |
|---|---|---|
| 位 15 ~ 12 | AK = 任务或应答识别标记 ID | 参看下文 |
| 位 11 | SPM = 参数修改报告 | 不支持（总是 0） |
| 位 10 ~ 00 | b. PNU = 基本参数号 | 完整的 PNU 由基本参数号与 IND 的 15 ~ 12 一起构成 |

**表 11-12　第二个字 IND**

| 第 2 个字 16 位 = IND = 参数的下标 | | |
|---|---|---|
| 位 15141312 ($2^0 2^3 2^2 2^1$) | PNU 扩展 PNU 页号 | 参看下文 |
| 位 11 ~ 10 | 备用 | 未使用 |
| 位 09 ~ 08 | 选择文本的类型 + 文本的读或写 | 未使用 |
| 位 07 ~ 00 | 下标哪个元素<br>哪个参数值<br>哪个元素说明<br>哪个下标文本是有效的<br>哪个数值文本是有效的 | 数值 255 = 下表参数数值或参数说明的全部元素。只有当 P2013 = 127 时才有可能 |

完整的参数号是由参数的任务/应答识别 ID（位 0 ~ 10）中的基本参数号和下标 PNU 页号中的位 12 ~ 15 一起产生的。第二个字 IND 参数下标见表 11-13。

**表 11-13　第二个字 IND 参数下标**

| 基本参数号任务/应答识别标记 ID 中的位 10 ~ 0 | PNU 页（下标中的位 15 ~ 12） | 完整的 PNU = 基本 PNU + PNU（页号 * 2000） |
|---|---|---|
| 0...1999 | 0 | 0...1999 |
| 0...1999 | 1 | 2000...3999 |
| 0...1999 | 2 | 4000...5999 |
| … | … | … |
| 0...1999 | 15 | 30000...31999 |

第三和第四个字，PWE1 和 PWE2 是被访问参数的数值，见表 11-14 和表 11-15。MICROMASTER4 的参数数值有许多不同的类型；整数（单字长或双字长）、十进制数（以 IEEE 浮点数的形式给出永远是双字长）以及下标参数（这里称为数组）。参数的含义决定于参数数值的类型和 P2013 的设置。

**表 11-14　第三个字 PWE1**

| 第三个字 PWE1 = 第一个参数数值 | | |
|---|---|---|
| | 参数数值的类型 | P2013 的设置 |
| 位 15 ~ 0 | = 对于非数组参数是参数的数值<br>= 对于数组参数是第 n 个参数的数值和对于第 n 个元素的任务 | 当 P2013 的值 = 3（固定长度为 3 个字）或 = 127（长度可变）以及单字长参数时 |
| | = 对于数组参数是第 1 个参数的数值和对于所有元素的任务 | 当 P2013 的值 = 127（长度可变）以及单字长参数时 |

（续）

| 第三个字 PWE1 = 第一个参数数值 | | |
|---|---|---|
| | =0 | 当 P2013 的值 =4（固定长度为 4 个字）以及单字长参数时 |
| | =参数数值的高位字（非数组参数）<br>=对于数组参数是参数数值的高位字和对于第 n 个元素任务的高位字 | 当 P2013 的值 =4（固定长度为 4 个字）或 =127（长度可变）以及双字长参数时 |
| | =对于数组参数是第一个参数数值的高位字和对于所有元素任务的高位字 | 当 P2013 的值 =127（长度可变）以及双字长参数时 |
| | 错误的数值 | 从站向主站发送，且应答标记 ID = 任务不能执行时 |

**表 11-15　第四个字 PWE2**

| 第四个字 PWE2 = 第二个参数数值 | | |
|---|---|---|
| | 参数数值的类型 | P2013 的设置 |
| 位 15~0 | =对于数组参数是第 2 个参数数值和对于所有元素的任务 | 当 P2013 的值 =4（固定长度为 4 个字）或 =127（长度可变）以及单字长参数时 |
| | =参数数值的低位字非数组参数<br>=对于数组参数是第 n 个参数数值的低位字和对于第 n 个元素任务的低位字 | 当 P2013 的值 =4（固定长度为 4 个字）或 =127（长度可变）以及双字长参数时 |
| | =对于数组参数是第 1 个参数数值的低位字和对于所有元素任务的低位字 | 当 P2013 的值 =127（长度可变）<br>以及双字长参数时 |
| | =下一个要访问的识别符标记 ID | 从站向主站传送，且应答识别标记 ID = 任务不能执行时。错误的数值 = ID 不存在或 ID 不能访问时<br>当 P2013 的值 =127 长度可变时 |
| | =下一个或前一个有效的数值 16 位<br>=下一个或前一个有效的数值 32 位的高位字<br>根据以下判定条件：<br>如果新值 > 实际值 向下一个有效的数值<br>如果新值 < 实际值向前一个有效的数值 | 从站向主站传送且应答识别标记 ID = 任务不能执行时。错误的数值 = 数值不可接受或有新的最大/最小值存在。当 P2013 的值 =127 长度可变时 |

参数识别标记 ID（PEK）总是一个 16 的值，位 0~10（PNU）包括所请求的参数号码，位 11（SPM）用于参数变更报告的触发位，位 12~15（AK）包括任务识别标记 ID 见表 11-16，应答标记 ID 见表 11-17。

**表 11-16　任务识别标记 ID**

| 任务识别标记 ID | 含　义 | 识别标记 ID | |
|---|---|---|---|
| | | 正 | 负 |
| 0 | 没有任务 | 0 | - |
| 1 | 请求参数数值 | 1 或 2 | 7 |
| 2 | 修改参数数值（单字）[只是修改 RAM] | 1 | 7 或 8 |
| 3 | 修改参数数值（双字）[只是修改 RAM] | 2 | 7 或 8 |
| 4 | 请求元素说明 | 3 | 7 |
| 5 | 修改元素说明（MICROMASTER4 不可能） | - | - |
| 6 | 请求参数数值（数组），即带下标的参数 | 4 或 5 | 7 |
| 7 | 修改参数数值（数组，单字）[只是修改 RAM] | 4 | 7 或 8 |

（续）

| 任务识别标记 ID | 含　　义 | 识别标记 ID | |
|---|---|---|---|
| | | 正 | 负 |
| 8 | 修改参数数值（数组，双字）[只是修改 RAM] | 5 | 7 或 8 |
| 9 | 请求数组元素的序号，即下标的序号，“no.” | 6 | 7 |
| 10 | 保留备用 | – | – |
| 11 | 存储参数数值（数组，双字）[RAM 和 EEPROM 都修改] | 5 | 7 或 8 |
| 12 | 存储参数数值（数组，单字）[RAM 和 EEPROM 都修改] | 4 | 7 或 8 |
| 13 | 存储参数数值（双字）[RAM 和 EEPROM 都修改] | 2 | 7 或 8 |
| 14 | 存储参数数值（单字）[RAM 和 EEPROM 都修改] | 1 | 7 或 8 |
| 15 | 读出或修改文本（MICROMASTER440 不可能） | – | – |

**表 11-17　应答识别标记 ID**

| 任务识别标记 ID | 含　　义 | 对任务识别标记 ID |
|---|---|---|
| 0 | 不应答 | 0 |
| 1 | 传送参数数值（单）字 | 1、2 或 14 |
| 2 | 传送参数数值（双字） | 1、3 或 13 |
| 3 | 传送说明元素 | 4 |
| 4 | 传送参数数值（数组，单字） | 6、7 或 12 |
| 5 | 传送参数数值（数组，双字） | 6、8 或 11 |
| 6 | 传送数组元素的数目 | 9 |
| 7 | 任务不能执行（有错误的数值） | 1 ~ 15 |
| 8 | 对参数接口没有修改权 | 2、3、5、7、8 |
| 9 – 12 | 未使用 | – |
| 13 | 预留，备用 | – |
| 14 | 预留，备用 | – |
| 15 | 传送文本 | 15 |

## 11.5.2　实例

**【例 11-7】**有一台设备，由 CPU314C – 2DP 控制一台 MM440 变频器，CPU314C – 2DP 通过 PROFIBUS – DP 控制 MM440 变频器，实现电动机的“正 – 停 – 反”和无级调速，频率数值在 HMI 中设定，起动系统后，默认的频率为 40 Hz。已知电动机的技术参数，功率为 0.75 kW，额定转速为 1400 r/min，额定电压为 380 V，额定电流为 2.05 A，额定频率为 50 Hz。请设计此系统，并编写程序。

**解：**

**1. 软硬件配置**

① 1 套 STEP 7 V5.5 SP4。

② 1 台 MM440 变频器（含 PROFIBUS 模板）。

③ 1 台 CPU 314C – 2DP。

④ 1 台电动机。

⑤ 1 根编程电缆。

⑥ 1 根 PROFIBUS 屏蔽双绞线。

⑦ 1 台 HMI。

硬件配置如图 11-15 所示，接线图如图 11-16。

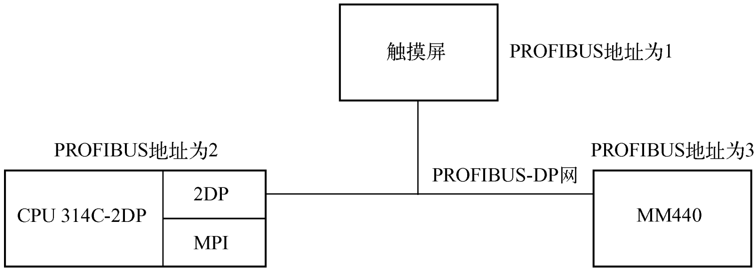



图 11-15　硬件配置图

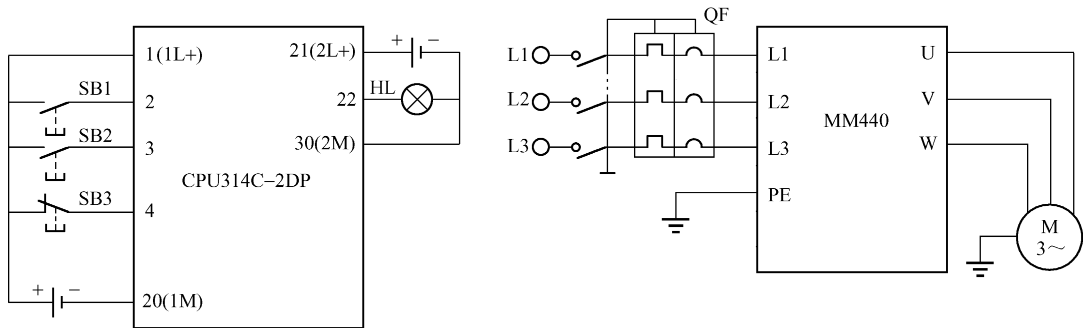



图 11-16　接线图

**2. MM440 变频器的设置**

MM440 变频器的参数见表 11-18。

**表 11-18　变频器参数表**

| 序号 | 变频器参数 | 出　厂　值 | 设　定　值 | 功 能 说 明 |
|---|---|---|---|---|
| 1 | P0304 | 380 | 380 | 电动机的额定电压（380 V） |
| 2 | P0305 | 3. 25 | 2. 05 | 电动机的额定电流（2. 05 A） |
| 3 | P0307 | 0. 75 | 0. 75 | 电动机的额定功率（0. 75 kW） |
| 4 | P0310 | 50. 00 | 50. 00 | 电动机的额定频率（50 Hz） |
| 5 | P0311 | 1400 | 1400 | 电动机的额定转速（1400 r/min） |
| 6 | P0700 | 2 | 6 | 选择命令源（COM 链路的通信板 CB 设置） |
| 7 | P1000 | 2 | 6 | 频率源（COM 链路的通信板 CB 设置） |
| 8 | P2009 | 0 | 1 | USS 规格化 |

MM440 变频器 PROFIBUS 站地址的设定在变频器的通信板（CB）上完成，通信板（CB）上有一排拨钮用于设置地址，每个拨钮对应一个“8－4－2－1”码的数据，所有的拨钮处于“ON”位置时，对应的数据之和就是站地址。拨钮示意图如图 11-17 所示，拨钮 1 和 2 处于“ON”位置，所以对应的数据为 1 和 2；而拨钮 3、4、5 和 6 处于“OFF”位置，所对应的数据为 0，站地址为 1＋2＋0＋0＋0＋0＋0＝3。




图 11-17　拨钮示意图

【关键点】图 11-17 设置的站地址为 3，必须和 STEP 7 软件中硬件组态的地址保持一致，否则不能通信。

### 3. S7 - 300 的硬件组态

1）新建项目和 PROFIBUS 网络。将项目命名为“通信速度给定”。新建“PROFIBUS”网络，设置 CPU314C - 2DP 的站地址为“2”，选中如图 11-18 所示中“1”处的网络，展开“PROFIBUS DP”。

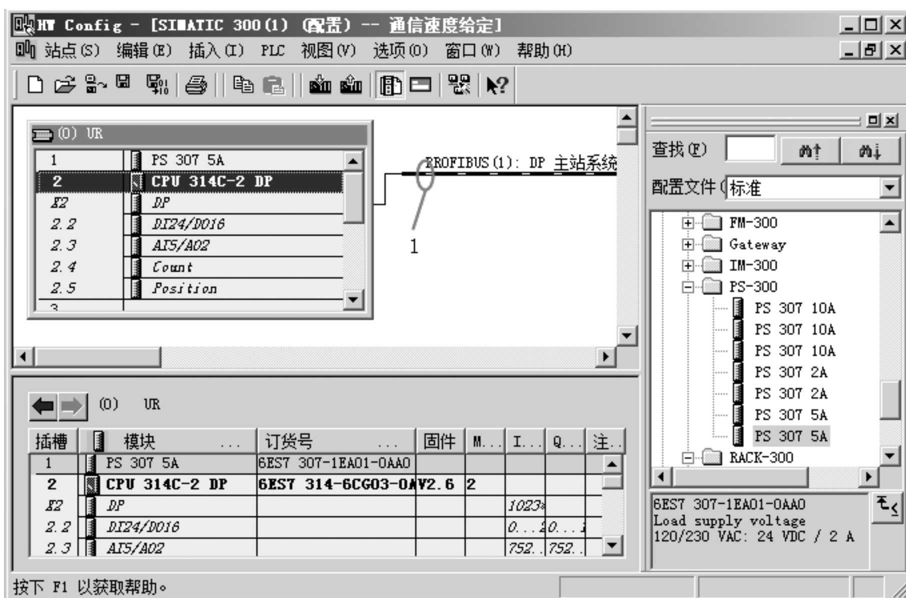



图 11-18　新建项目和 PROFIBUS 网络

2）选中“MICROMASTER 4”。如图 11-19 所示，先展开“SIMOVERT”，再选中“MICROMASTER 4”，并双击之，弹出如图 11-20 所示的界面。

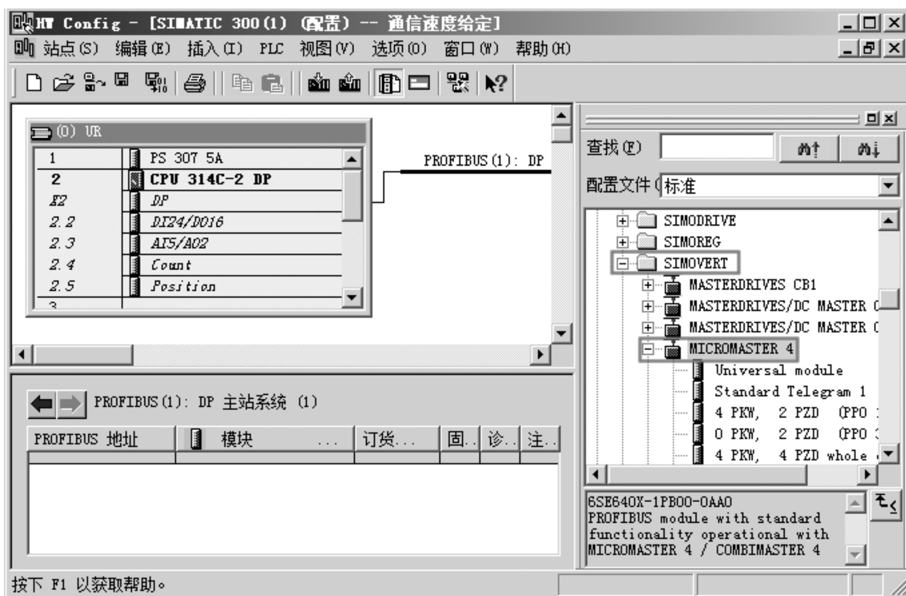



图 11-19　选中“MICROMASTER 4”

3）设置 MM440 的站地址。如图 11-20 所示，先选中“PROFIBUS(1)”网络，再将“地址”设置为“3”，最后单击“确定”按钮。




图 11-20　设置 MM440 的站地址

4）选择通信报文的结构。PROFIBUS 的通信报文由两部分组成，即 PKW（参数识别 ID 数据区）和 PZD 区（过程数据）。如图 11-21 所示，先选中“1”处，再双击“0 PKW,2 PZD（PPO3）”，“0 PKW,2 PZD（PPO3）”通信报文格式的含义是：报文中没有 PKW，只有 2 个字的 PZD。

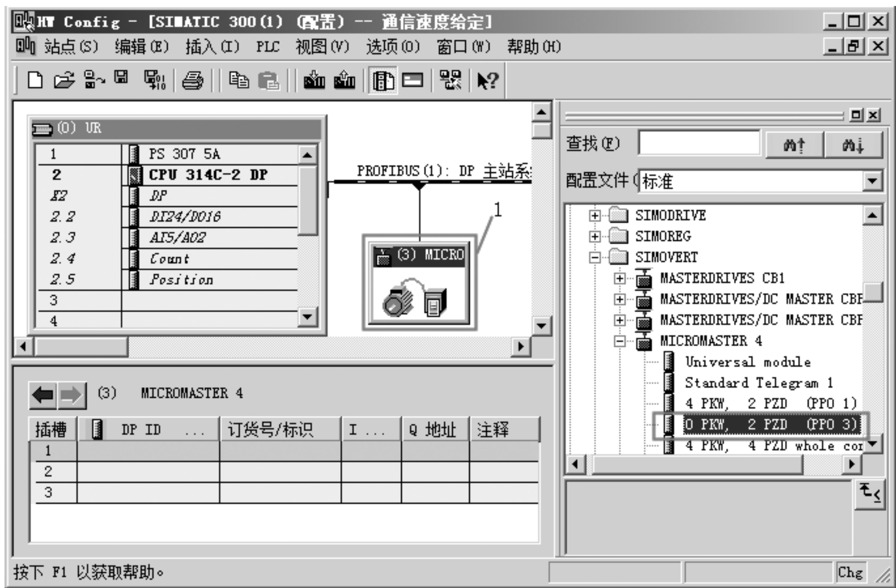



图 11-21　选择通信报文的结构

5）MM440 的数据地址。如图 11-22 中，S7-300（主站）发送到 MM440 的数据存放在 QB256～QB259（共两个字），S7-300（主站）接收 MM440 的数据区在 IB256～IB259（共两个字）。最后，保存和编译组态完成的硬件。

**【关键点】**“发送”和“接收”都是对主站而言的。

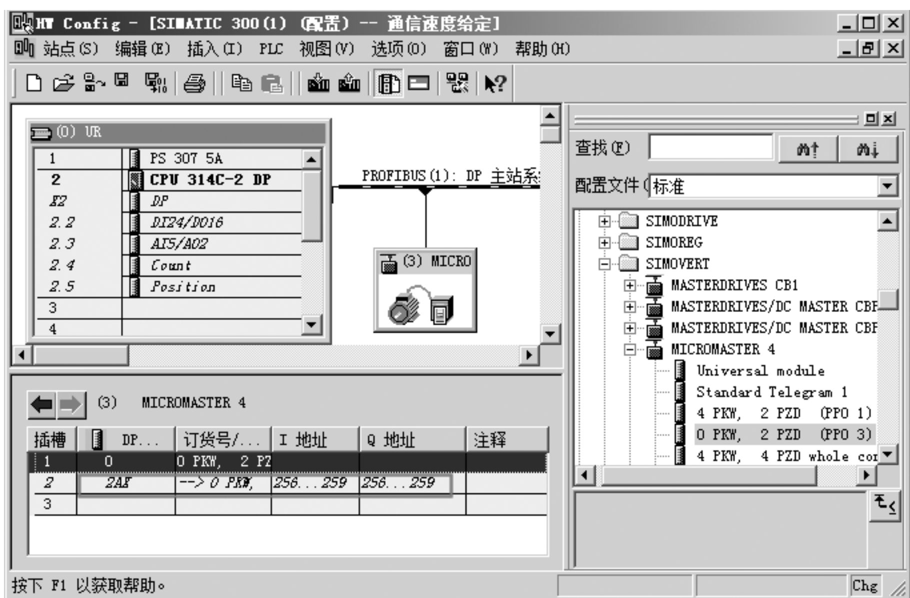


图 11-22　MM440 的数据地址

**4. 编写程序**

（1）任务报文 PZD 的介绍

任务报文的 PZD 区是为控制和检测变频器而设计的。PZD 的第一个字是变频器的控制字（STW）。变频器的 STW 控制字见表 11-19。

**表 11-19　变频器的 STW 控制字**

| 位 | 项　　目 | 含　　义 |
|---|---|---|
| 位 00 | ON（斜坡上升）/OFF1（斜坡下降） | 0 否，1 是 |
| 位 01 | OFF2：按照惯性自由停车 | 0 是，1 否 |
| 位 02 | OFF3：快速停车 | 0 是，1 否 |
| 位 03 | 脉冲使能 | 0 否，1 是 |
| 位 04 | 斜坡函数发生器（RFG）使能 | 0 否，1 是 |
| 位 05 | RFG 开始 | 0 否，1 是 |
| 位 06 | 设定值使能 | 0 否，1 是 |
| 位 07 | 故障确认 | 0 否，1 是 |
| 位 08 | 正向点动 | 0 否，1 是 |
| 位 09 | 反向点动 | 0 否，1 是 |
| 位 10 | 由 PLC 进行控制 | 0 否，1 是 |
| 位 11 | 设定值反向 | 0 否，1 是 |
| 位 12 | 未使用 | |
| 位 13 | 用电动电位计 MOP 升速 | 0 否，1 是 |
| 位 14 | 用电动电位计 MOP 降速 | 0 否，1 是 |
| 位 15 | 本机/远程控制 | 0P7019 下标 0，1P7019 下标 1 |

PZD 的第二个字是变频器的主设定值（HSW），也是主频率设定值。有两种不同的设置方式，当 P2009 设置为 0 时，数值以十六进制形式发送，即 4000（hex）规格化为由 P2000

（默认值为 50）设定的频率，4000 相当于 50 Hz。当 P2009 设置为 1 时，数值以十进制形式发送，即 4000（十进制）表示频率为 40.00 Hz。

例如当 P2009 = 0 时，任务报文为 PZD = 047F4000，第一个字的二进制为 0000，0100，0111，1111。这个字的含义是：斜坡上升；不是自由惯性停机；不是快速停车；脉冲使能；斜坡函数发生器（RFG）使能；RFG 开始；设定值使能；不确认故障；不是正向点动；不是反向点动；PLC 进行控制；设定值不反向；不用 MOP 升速和降速。第二个字的含义是：转速为 50Hz。

（2）频率的计算方法

以下用一个例子介绍变频器通信时，规格化的频率计算方法。

**【例 11-8】**变频器通信时，需要对频率进行规格化，计算 40 Hz 对应的规格化数值。

**解：**因为 50 Hz 对应的十六进制是 4000，而 16#4000 对应的十进制是 16384，所以 40 Hz 对应的十进制是：

$$f = \frac{40.0}{50.0} \times 16384 = 13107.2$$

而 13107 对应的十六进制是 16#3333，所以设置时，应设置数值是 16#3333。初学者容易用 4000 × 0.8 = 3200，显然不对。

（3）应答报文 PZD 的介绍

应答报文 PZD 的第一个字是变频器的状态字（ZWS）。变频器的状态字通常由参数 r0052 定义。变频器的状态字（ZSW）含义见表 11-20。

**表 11-20　变频器的状态字 ZSW**

| 位 | 项　目 | 含　义 | 位 | 项　目 | 含　义 |
|---|---|---|---|---|---|
| 位 00 | 变频器准备 | 0 否，1 是 | 位 08 | 设定值/实际偏差过大 | 0 是，1 否 |
| 位 01 | 变频器运行准备就绪 | 0 否，1 是 | 位 09 | 过程数据监控 | 0 否，1 是 |
| 位 02 | 变频器正在运行 | 0 否，1 是 | 位 10 | 已经达到最大频率 | 0 否，1 是 |
| 位 03 | 变频器故障 | 0 是，1 否 | 位 11 | 电动机极限电流报警 | 0 是，1 否 |
| 位 04 | OFF2 命令激活 | 0 是，1 否 | 位 12 | 电动机抱闸制动投入 | 0 是，1 否 |
| 位 05 | OFF3 命令激活 | 0 否，1 是 | 位 13 | 电动机过载 | 0 是，1 否 |
| 位 06 | 禁止接通 | 0 否，1 是 | 位 14 | 电动机正向运行 | 0 否，1 是 |
| 位 07 | 变频器报警 | 0 否，1 是 | 位 15 | 变频器过载 | 0 是，1 否 |

应答报文的 PZD 第二个字是变频器的运行实际参数（HIW）。通常定义为变频器的实际输出频率。其数值也由 P2009 进行规格化。

（4）编写程序

编写程序如图 11-23 和图 11-24 所示。

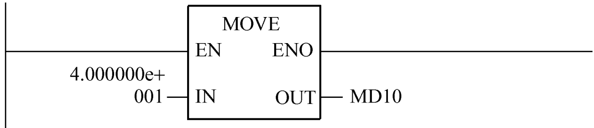
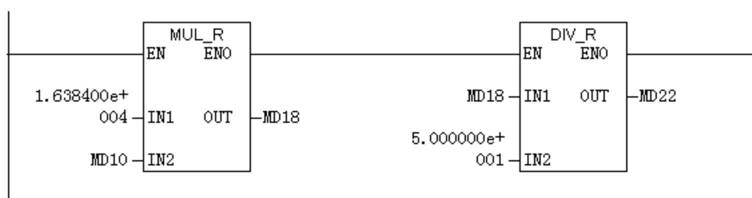


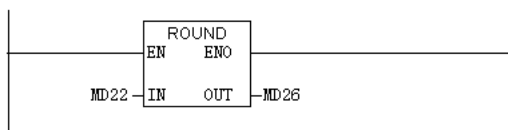

图 11-23　OB100 中的程序

**程序段 1**：标题：

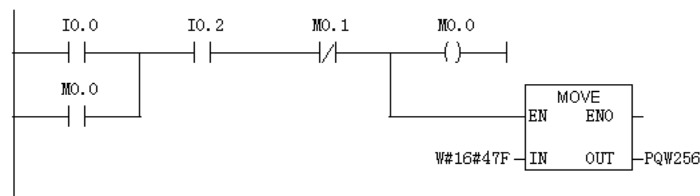



**程序段 2**：标题：

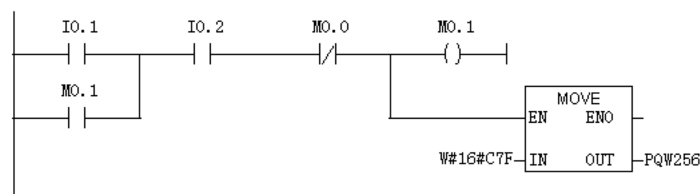
MW28中存放的就是规格化后的频率值



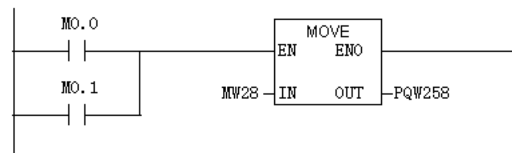

**程序段 3**：正转控制




**程序段 4**：反转控制




**程序段 5**：运行时，频率给定




**程序段 6**：停止时，频率给定和发送停止命令

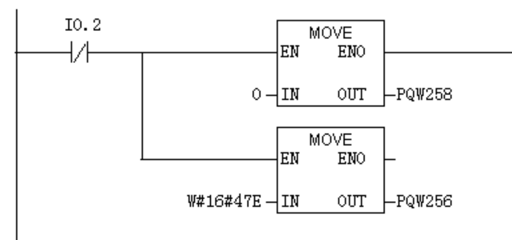



图 11-24　OB1 中的程序

### 11.5.3 S7 -300 通过 PROFIBUS 现场总线修改 MM440 变频器的参数

S7 -300 通过 PROFIBUS 现场总线通信修改变频器的参数，实际上就是 S7 -300 和 MM440 变频器现场总线通信的一种。以下用一个例子介绍 S7 -300 通过 PROFIBUS 现场总线修改 MM440 变频器的参数。

**【例 11-9】** 利用一台 CPU314C -2DP 通过 PROFIBUS 现场总线通信修改 MM440 变频器的参数，将 P701 原有数值 1 修改成 2，请设计方案，并编写程序。

**解：**

**1. 软硬件配置**

① 1 套 STEP 7 V5.5 SP4。

② 1 台 MM440 变频器（含 PROFIBUS 模板）。

③ 1 台 CPU314C -2DP。

④ 1 根 PROFIBUS 屏蔽双绞线。

⑤ 1 根编程电缆。

硬件配置方案如图 11-25 所示。




图 11-25 硬件配置方案

**2. 硬件组态**

1）新建项目和 PROFIBUS 网络。如图 11-26 所示，将项目命名为“profibusWrite”。新建 PROFIBUS 网络，设置 CPU314C -2DP 的站地址为 2，选中如图 11-27 所示中“1”处的网络，展开“PROFIBUS DP”。




图 11-26 新建项目和 PROFIBUS 网络

2）选中“MICROMASTER 4”。如图 11-27 所示，先展开“SIMOVERT”，再选中“MICROMASTER 4”，并双击之，弹出图 11-28 所示的界面。

3）设置 MM440 的站地址。如图 11-29 所示，先选中“PROFIBUS(1)”网络，再将“地址”设置为 5，最后单击“OK”（确认）按钮。

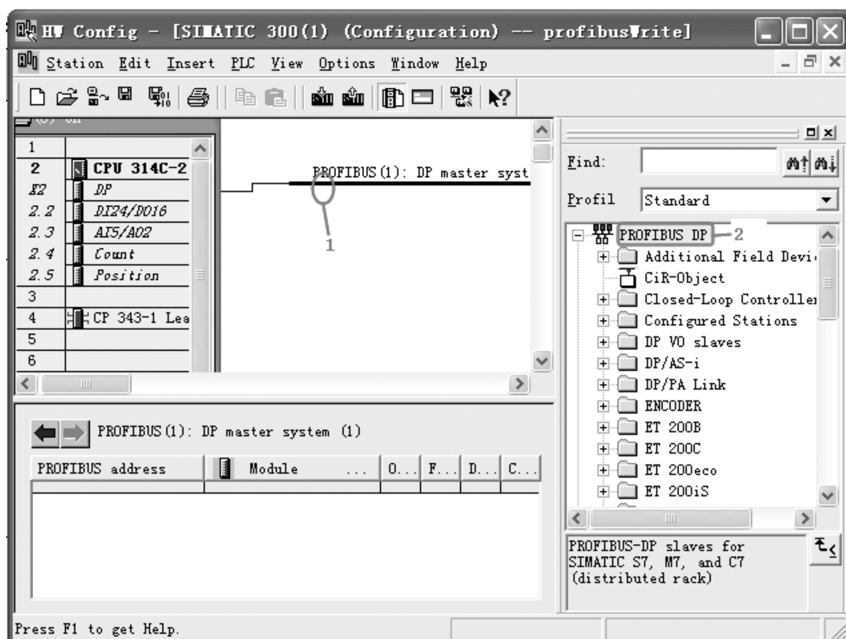



图 11-27　展开“PROFIBUS DP”

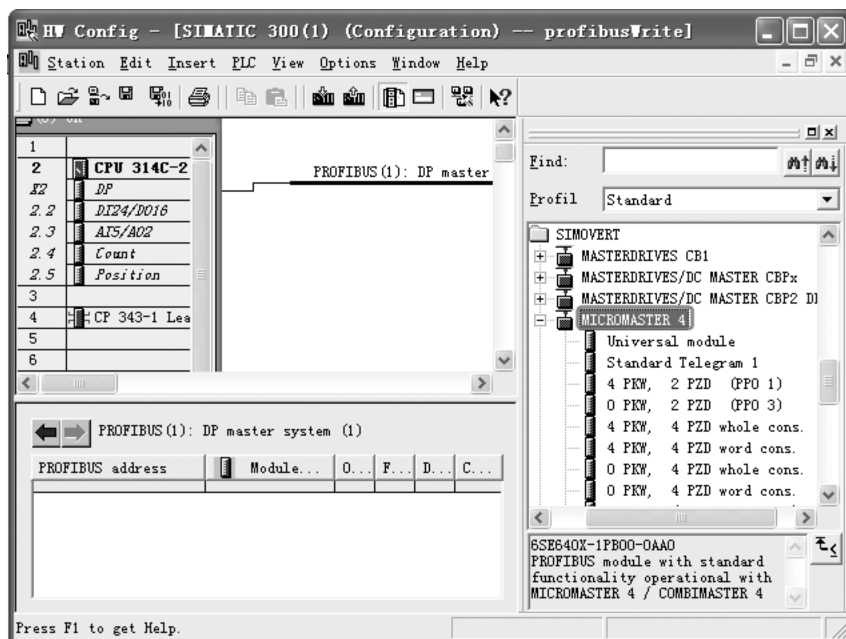



图 11-28　选中“MICROMASTER 4”

4）选择通信报文的结构。PROFIBUS 的通信报文由两部分组成，即 PKW（参数识别 ID 数据区）和 PZD 区（过程数据）。如图 11-30 所示，先选中“1”处，再双击“4 PKW，2 PZD（PPO1）”，“4 PKW，2 PZD（PPO1）”通信报文格式的含义是报文中有四个字的 PKW，有两个字的 PZD。

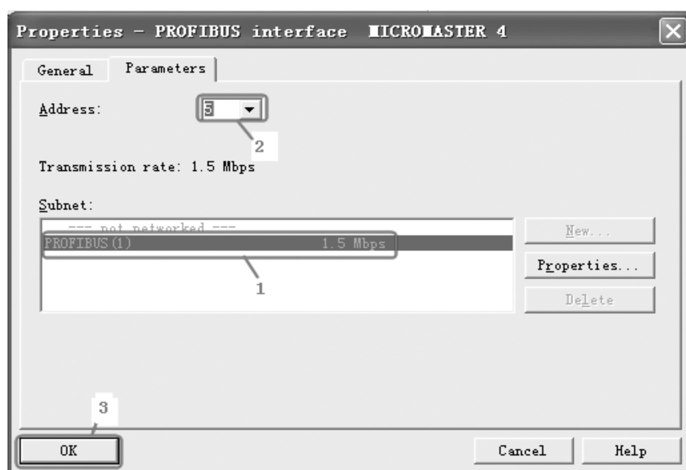



图 11-29　MM440 的站地址

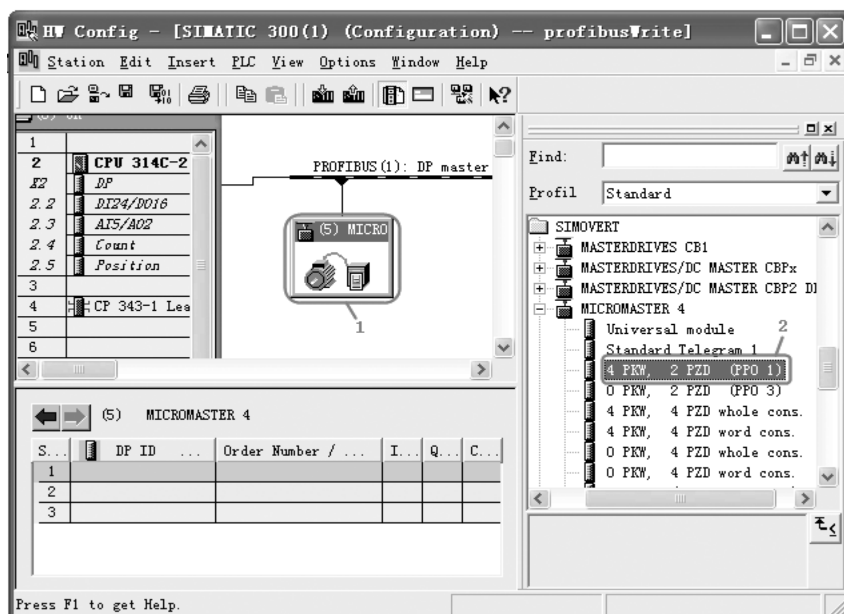



图 11-30　选择通信报文的结构

5）MM440 的数据地址。如图 11-31 中，主站发送给 MM440 的 PKW 数据存放在 QB256 ~ QB263（共四个字），主站接收 MM440 反馈信息的数据区在 IB256 ~ IB263（共四个字）。而主站发送给 MM440 的 PZD 数据存放在 QB264 ~ QB267（共两个字），主站接收 MM440 反馈信息的数据区在 IB264 ~ IB267（共两个字）。

6）插入组织块、数据块和参数表。返回 SIMATIC Manager 界面，插入组织块 OB80、OB82、OB84、OB87、OB121 和 OB122，再插入数据块 DB1 和参数表 VAT_1，如图 11-32 所示。

7）在数据块中创建数组。双击如图 11-32 中的数据块“DB1”，创建数组“DB_VAR”，如图 11-33 所示。

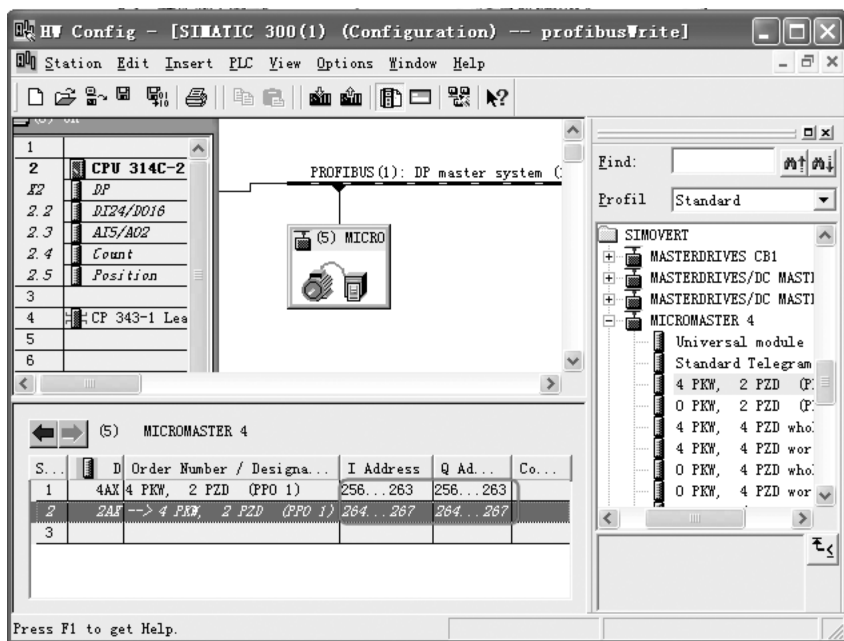



图 11-31 MM440 的数据地址

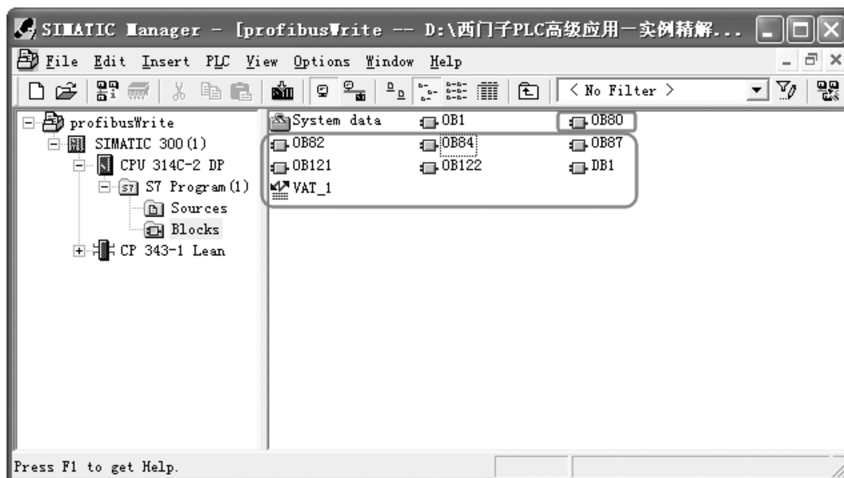



图 11-32 插入组织块、数据块和参数表

8）在参数表中输入 PKW 参数。双击如图 11-32 中的参数表“VAT_1”，弹出参数表，如图 11-34 所示。“1”处的参数为写入 MM440 变频器的参数，“2”处的参数为向变频器读写参数的开关。

**3. 相关指令简介**

在组态接收和发送时，经常遇到“Consistency”（一致性），当选择“Unit”时，则以字节发送和接收数据。如果数据到达从站接收区不在同一时刻，从站可能不能在同一周期处理完接收区数据。如果需要从站必须在同一周期内处理完这些数据，可选择“All”选项，编程时调用 DPWR_DAT 打包发送，从 DP 从站或者 PROFINET IO 设备上发送连续

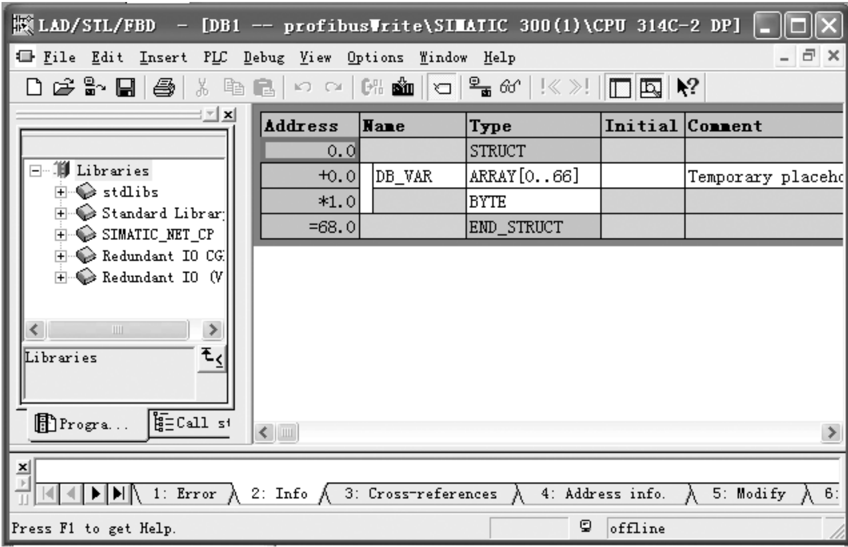



图 11-33　在数据块中创建数组

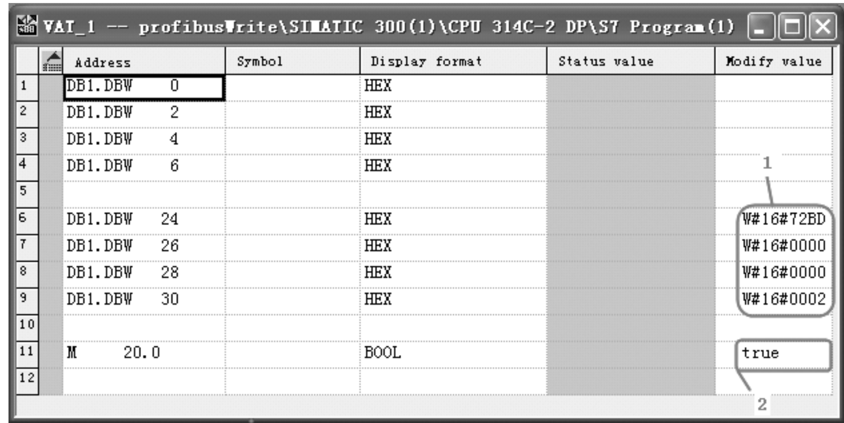



图 11-34　在参数表中输入 PKW 参数

数据，调用 DPRD_DAT 解包接收，从 DP 从站或者 PROFINET IO 设备上接收连续数据。打包发送（DPWR_DAT）的指令格式见表 11-21，打包接收（DPWR_DAT）的指令格式见表 11-22。

**表 11-21　DPRD_DAT（SFC 14）指令格式**

| LAD | 输入/输出 | 含　　义 | 数 据 类 型 |
|---|---|---|---|
| "DPRD_DAT"<br>EN　ENO<br>LADDR　RET_VAL<br>RECORD | EN | 使能 | BOOL |
| | LADDR | 对方数据起始地址，其实就是本地要接收的解包数据存放在对方的起始地址 | WORD |
| | RET_VAL | 返回值是错误代码 | INT |
| | RECORD | 接收解包数据后存放的地址 | ANY |

表 11-22　DPWR_DAT(SFC 15)指令格式

| LAD | 输入/输出 | 含　　义 | 数据类型 |
|---|---|---|---|
| “DPWR_DAT”<br>EN　ENO<br>LADDR　RET_VAL<br>RECORD | EN | 使能 | BOOL |
| | LADDR | 对方数据起始地址，其实就是对方要接收的数据存放的起始地址 | WORD |
| | RET_VAL | 返回值是错误代码 | INT |
| | RECORD | 本地要发送数据存放的地址 | ANY |

**4. 编写程序**

在编写程序前先对图 11-34 中参数表中的含义进行解释。

1）W#16#72BD，PKW 的第一个字，即参数识别标记 ID，显然是用十六进制表示，“7”表示修改参数数值，此参数为数组、单字（见表 11-12、表 11-14 和表 11-17），“2BD”就是 701 的十六进制。

2）W#16#0000，PKW 的第二个字，即参数下标（见表 11-12 和表 11-13），显然 701 小于 2000，所以其下标为 0。

3）W#16#0000，PKW 的第三个字，第一个参数值（PWE1），为 0（见表 11-12）。

4）W#16#0002，PKW 的第四个字，第二个参数值（PWE2），为 2，是要修改的新数值（见表 11-13）。

5）梯形图程序如图 11-35 所示。

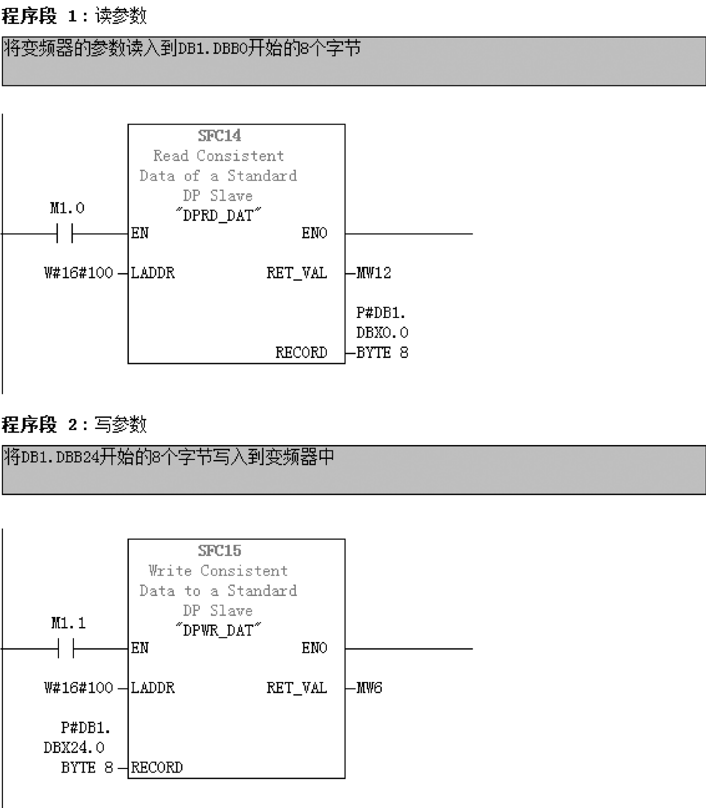



图 11-35　梯形图

【关键点】要编写正确的程序，首先必须理解 PKW 各个字的含义，其次要理解 DPRD_DAT 和 DPWR_DAT 的用法。如图 11-35 的程序中“LADDR”前的“W#16#100”是用十六进制表示的，与图 11-31 中的地址是对应的，这点必须注意，否则通信是不能成功的。

## 11.5.4 S7-300 通过 PROFIBUS 现场总线控制 S120

**1. STARTER 软件概述**

STARTER 软件用于西门子传动装置的现场调试。能够实现在线监控、修改装置参数、故障检测和复位以及跟踪记录等强大的调试功能。

**2. SINAMICS S120 AC/AC 单轴驱动器概述**

SINAMICS S120 AC/AC 单轴驱动器是西门子公司推出的新一代交流驱动产品—集整流和逆变于一体的新型驱动器，既能实现通常的 U/f、矢量控制，又能实现高准确度、高性能的伺服控制功能。它不仅能控制普通的三相异步电动机，还能控制异步和同步伺服电动机、扭矩电动机及直线电动机。其强大的定位功能可以实现进给轴的绝对、相对定位。

SINAMICS S120 产品包括用于共直流母线的 DC/AC 逆变器和用于单轴的 AC/AC 变频器。共直流母线的 DC/AC 逆变器通常又称为 SINAMICS S120 多轴驱动器，其结构形式为电源模块和电动机模块分开，一个电源模块将三相交流电整流成 540 V 或 600 V 的直流电，将电动机模块（一个或多个）都连接到该直流母线上，特别适用于多轴控制，尤其是造纸、包装、纺织、印刷、钢铁等行业。其优点是各电动机轴之间的能量共享，接线方便、简单。单轴控制的 AC/AC 变频器，通常又称为 SINAMICS S120 单轴交流驱动器，其结构形式为电源模块和电动机模块集在一起，特别适用于单轴的速度和定位控制。

本部分只介绍 SINAMICS S120 单轴交流驱动器。SINAMICS S120 AC/AC 单轴驱动器由两部分组成：控制单元和功率模块。具体如下：

（1）控制单元有三种形式：CU310DP、CU310 PN 和 CUA31

CU310DP 是驱动器通过 PROFIBUS-DP 与上位的控制器相连。

CU310PN 是驱动器通过 PROFINET 与上位的控制器相连。

CUA31 是控制单元的适配器，通过 Drive-CLiQ 与 CU320 或 Simotion D 相连。

（2）功率模块有模块型和装机装柜型两种形式

模块型：其功率范围从 0.12～90 kW，其进线电压有单相（200～240 V）及三相（380～480 V）两种规格。

装机装柜型：其功率范围从 110～250 kW，其进线电压为三相（380～480 V）。

**3. 用 S7-300 控制 SINAMICS S120**

以下用一个例子介绍使用现场总线控制西门子伺服系统 SINAMICS S120。

**【例 11-10】**某设备上有一套 CU310DP 伺服驱动系统，要求：对伺服进行调速，请编写程序。

**【解】**

（1）软硬件配置

① 1 套 STEP 7 V5.5 SP4。

② 1 套 STARTER 4.4（或者 SCOUT）。

③ 1 套 CU310DP 伺服系统。

④ 1 台 CPU314C－2DP。

⑤ 1 根 PROFIBUS 屏蔽双绞线。

⑥ 1 块 CP5611 卡。

系统的硬件配置如图 11-36 所示，图中的 X22 口为 RS－232C 母头，是 SINAMICS S120 与计算机或者 PLC 通信的接口。

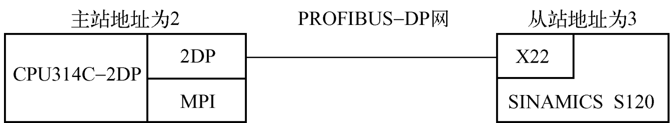



图 11-36　硬件配置图

（2）伺服系统参数的设置

① 设置伺服驱动器的站地址。本例伺服系统的站地址设置为 3。伺服驱动器上有一排拨钮用于设置地址，每个拨钮对应一个“8－4－2－1”码的数据，所有的拨钮处于“ON”位置时，对应的数据之和就是站地址。拨钮示意图如图 11-37 所示，拨钮 1 和 2 处于“ON”位置，所以对应的数据为 1 和 2；而拨钮 3、拨钮 4、拨钮 5 和拨钮 6 处于“OFF”位置，所对应的数据为 0，站地址为 1＋2＋0＋0＋0＋0＋0＝3。这个设置方法与 MM440 变频器的站地址设置是一样的，因为实际上 SINAMICS S120 也是变频器。

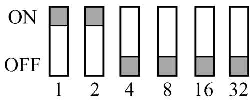



图 11-37　拨钮示意图

**【关键点】** 图 11-37 设置的站地址 3，必须和 STEP 7 软件中硬件组态的地址保持一致，否则不能通信。此外，设置完成后必须断电，设置的地址才能起作用。

② 打开 STARTER 软件，新建项目。在打开 STARTER 项目前，计算机中要首先安装 STARTER 软件，此软件不需要授权，可直接从西门子的网站上下载，若使用 SCOUT 则需要购买授权。打开 STARTER 软件，如图 11-38 所示，再单击工具栏上的“新建”按钮，弹出如图 11-39 所示的界面，将项目命名为“SERVER2”，最后单击“OK”按钮。

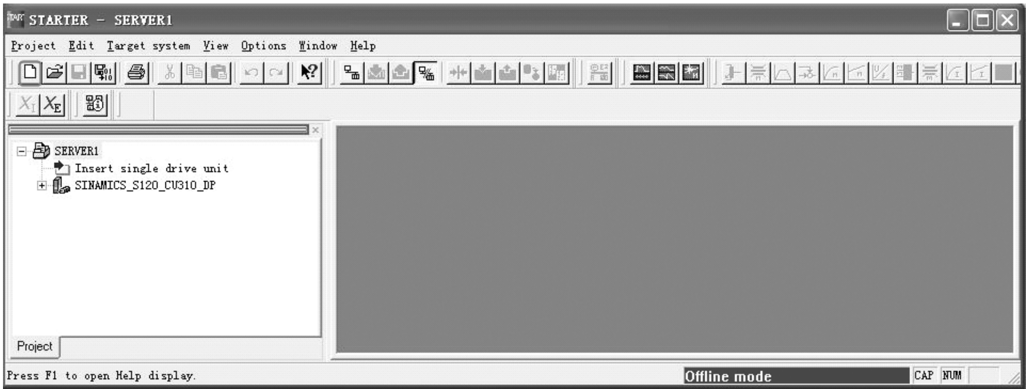



图 11-38　新建项目

③ 设置 PC 与 SINAMICS S120 通信的路径。先选中“SERVER2”，再在菜单中单击“Options”→“Set PG/PC Interface”，弹出“Set PG/PC Interface”界面，选中“CP5611（PROFIBUS）”选项，最后单击“OK”按钮，如图 11-40 所示。

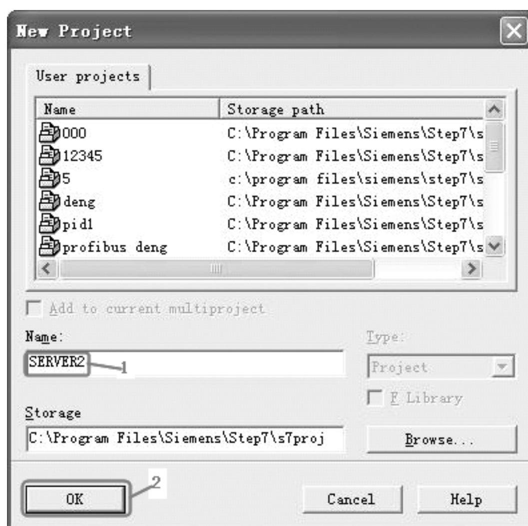



图 11-39　项目命名

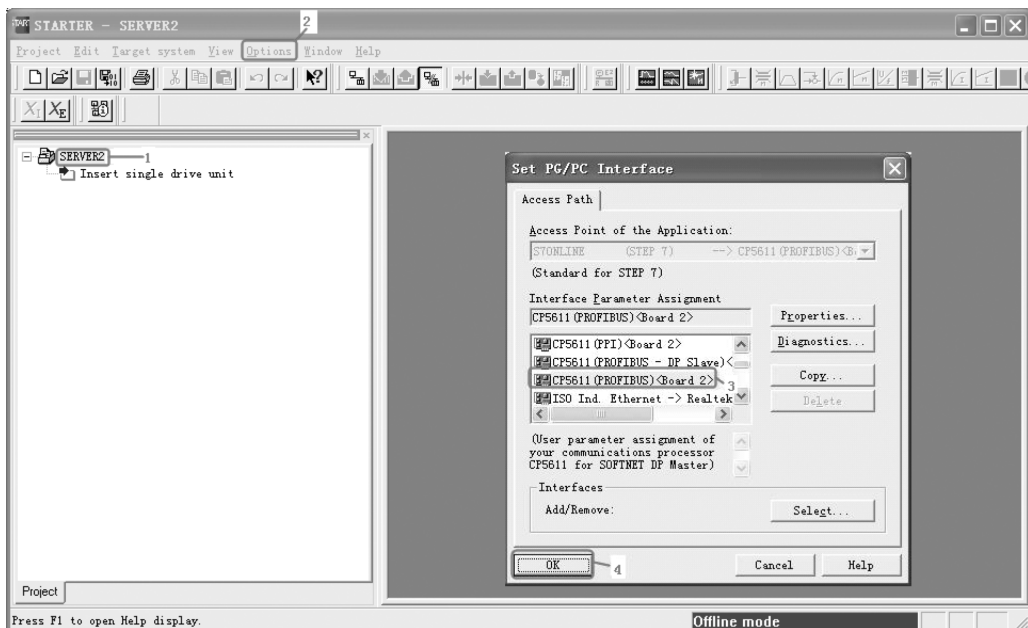



图 11-40　设置 PC 与 SINAMICS S120 通信的路径

④ 将 PG（编程器）与 SINAMICS S120 连接起来。先单击工具栏中的“connect to target system”（连接）按钮，再单击“Yes”按钮，STARTER 自动寻找可连线的目标，如图 11-41 所示。当 STARTER 找到目标时弹出如图 11-42 所示的界面，选中找到的目标（其地址为 3），该目标就是要连接的 SINAMICS S120，再单击“Accept”（接受）按钮。

⑤ 配置驱动器系统。驱动器系统参数的配置可以手动配置，也可以自动配置。手动配置较麻烦，因此推荐用自动配置。如图 11-43 所示，当将 PG 与 SINAMICS S120 连接起来后，可以看到“3”处的字是“Online mode”（在线模式），选中并双击“Automatic configu-

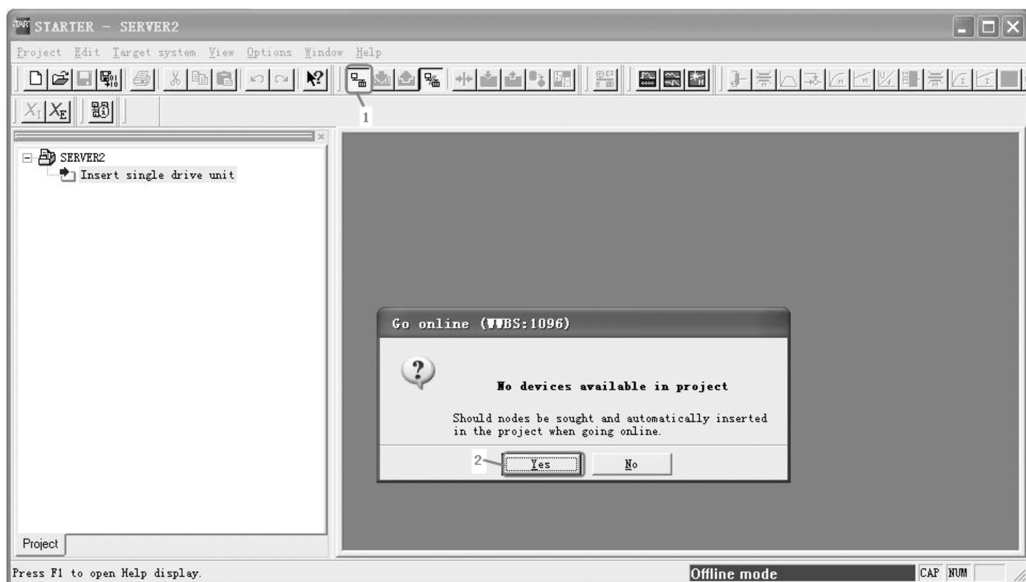



图 11-41　将 PC（编程器）与 SINAMICS S120 连接起来（1）

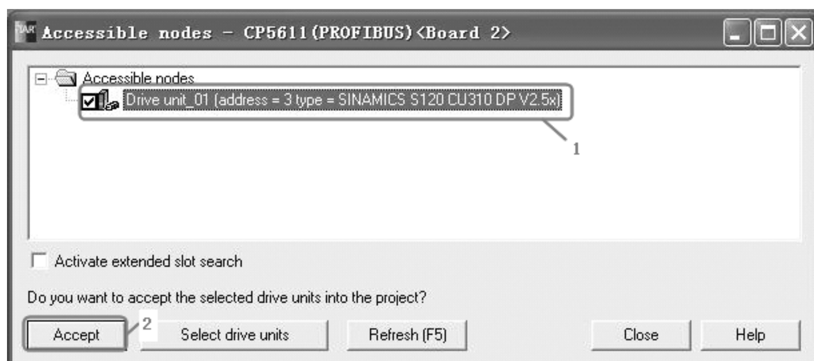



图 11-42　将 PC（编程器）与 SINAMICS S120 连接起来（2）

ration”（自动配置），再单击“Load to PG”（上载到编程器），弹出上载状态界面，如图 11-44 所示。当上载结束后，伺服系统的参数自动上传到编程器（PG）中。

⑥ 设置通信报文类型。先选中“PROFIBUS”，再选中“PROFIBUS message frame”选项卡，在“Message frame type”（通信报文类型），选定“Standard telegram 1”（通信报文 1），如图 11-45 所示。

⑦ 将 RAM 的数据复制到 ROM 中。先选中“Drive_unit_01”，再单击“Copy RAM to ROM”（将 RAM 的数据复制到 ROM）按钮，如图 11-46 所示。当数据复制完成后，再单击“Disconnect from target system”（离线）按钮，这样 PG 和 SINAMICS S120 通信连接断开。断电后，将两者的通信电缆拔下，再将 S7-300 的 DP 口与 SINAMICS S120 的 X22 口用屏蔽双绞线（含两只网络连接器）相连。

**【关键点】** 图 11-46 中的通信报文设置与 S7-300 PLC 中的报文设置要一致，否则通信不能成功。此外，必须将 RAM 的数据复制到 ROM 中，否则设置的报文就不起作用。

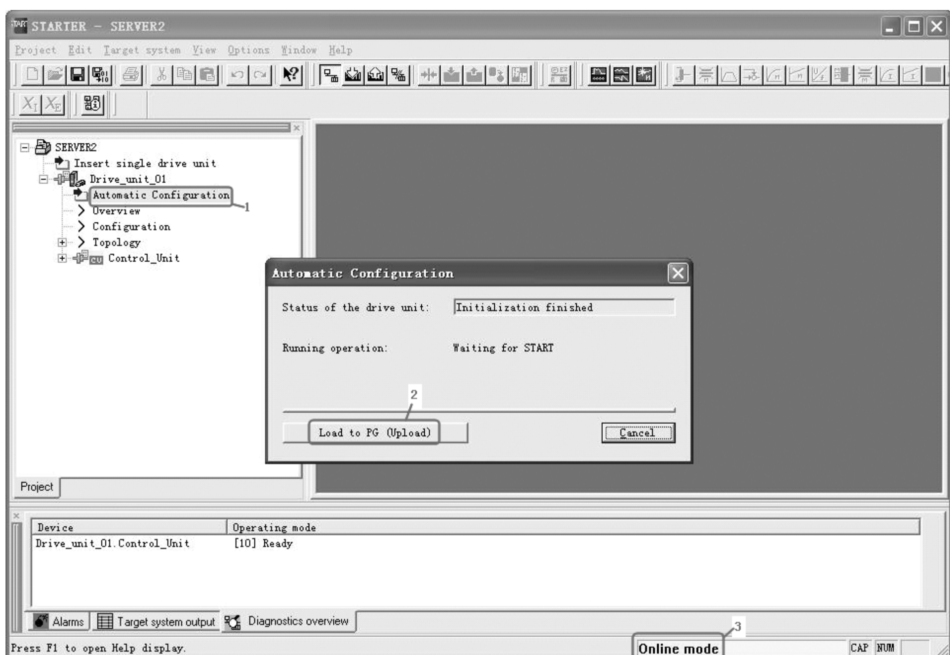



图 11-43　自动配置驱动器系统

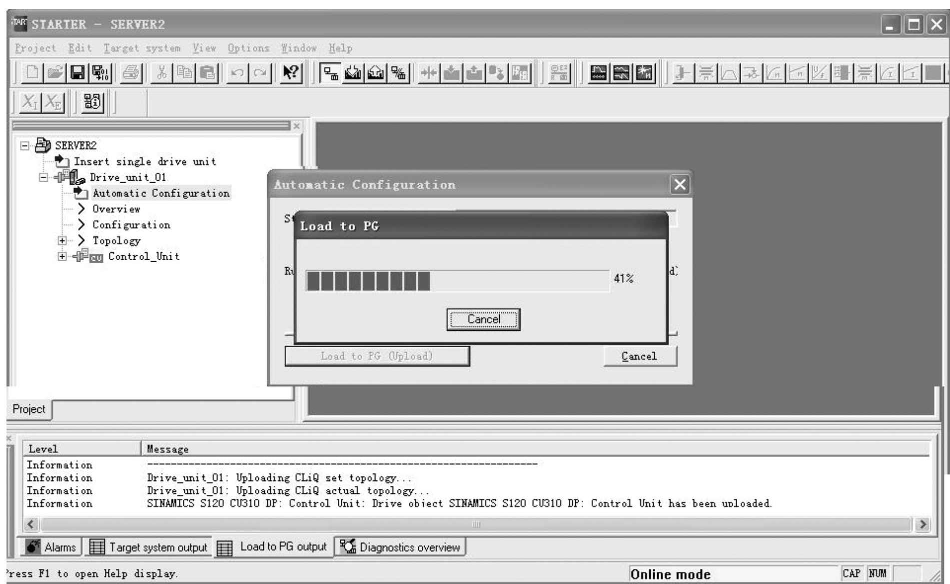



图 11-44　上载状态

（3）S7-300 PLC 的硬件组态

① 新建项目，并进行硬件组态。新建项目，命名为“314C-CU310-01”，如图 11-47 所示。

② 将 SINAMICS S120 挂到 PROFIBUS 总线上。将界面切换到硬件组态上，先选中“1”处，再双击“SINAMICS S”，如图 11-48 所示，SINAMICS S120 挂到 PROFIBUS 总线上。

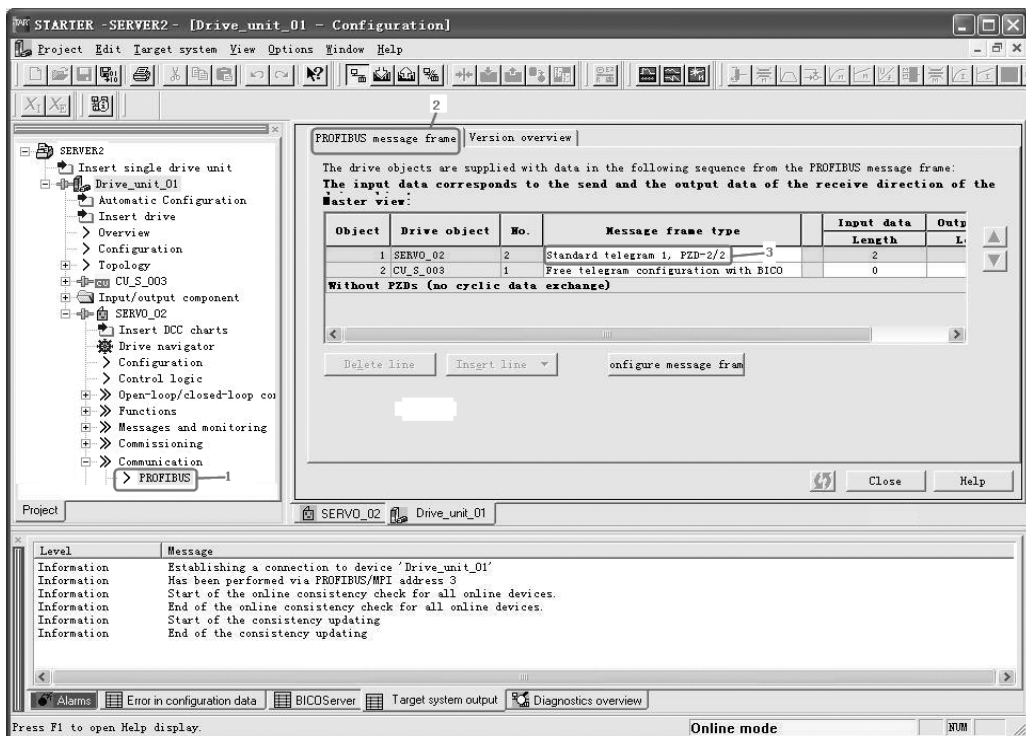



图 11-45　设置通信报文类型

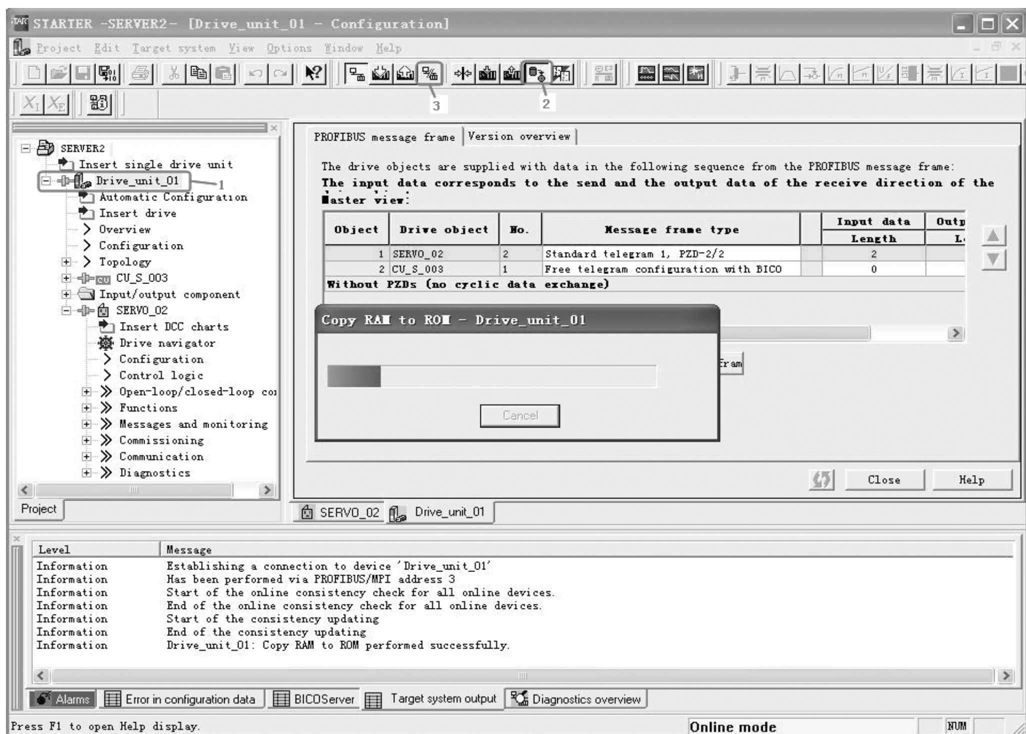



图 11-46　将 RAM 的数据复制到 ROM 中



图 11-47　新建项目

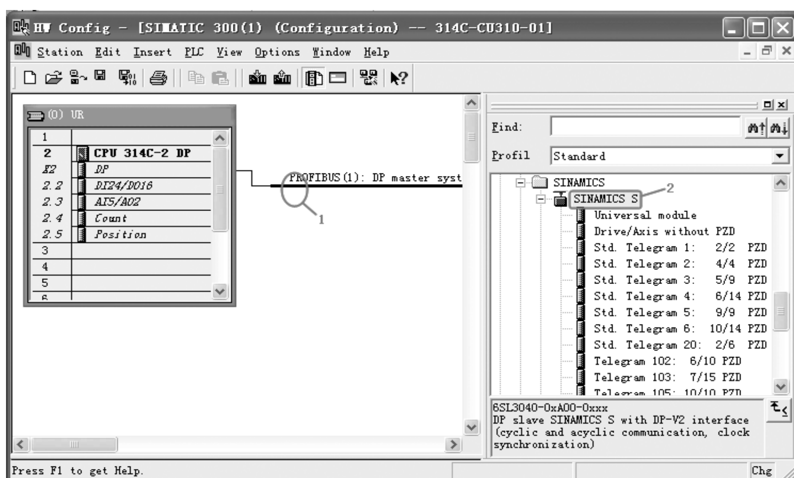


图 11-48　将 SINAMICS S120 挂到 PROFIBUS 总线上

③ 设置 SINAMICS S120 的站地址。先选中 PROFIBUS 网络，再设置 SINAMICS S120 的站地址为“3”。注意，这个地址与用拨码开关设置的 SINAMICS S120 的地址要一致，最后单击“OK”按钮，如图 11-49 所示。

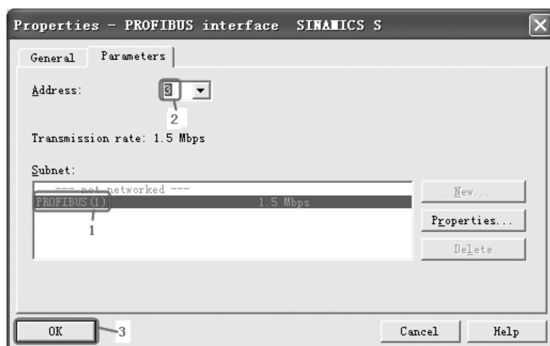


图 11-49　设置 SINAMICS S120 站地址

④ 设置 SINAMICS S120 的报文格式。先选中 SINAMICS S120，再双击“Std. Telegam 1：2/2 PZD”（报文 1），如图 11-50 所示，再保存和编译硬件组态。注意，此处的报文格式要与图 11-45 中的“3”处一致。

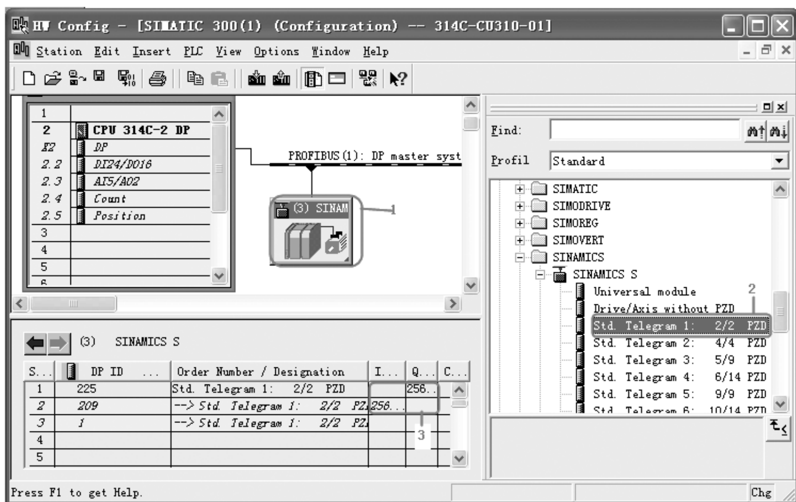



图 11-50　设置 SINAMICS S120 的报文格式

（4）编写程序

① 插入数据块 DB2 和参数表 VAT_1。选中块 OB1，插入数据块 DB2 和参数表 VAT_1，如图 11-51 所示。

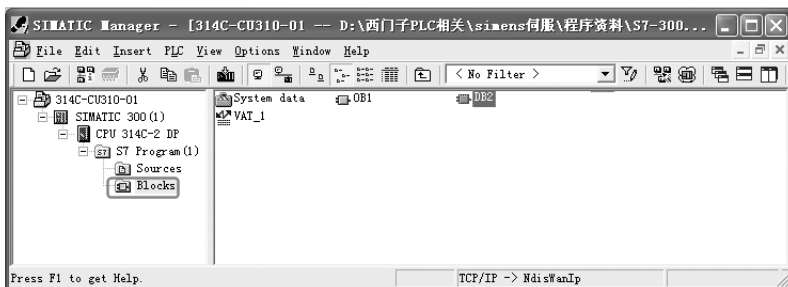



图 11-51　插入数据块 DB2 和参数表 VAT_1

② 在 DB2 中创建数组。打开数据块 DB2，并在 DB2 中创建数组 ARRAY[1..5]，如图 11-52 所示。用同样的方法创建数据块 DB1，并在 DB1 中创建数组 ARRAY[1..5]。

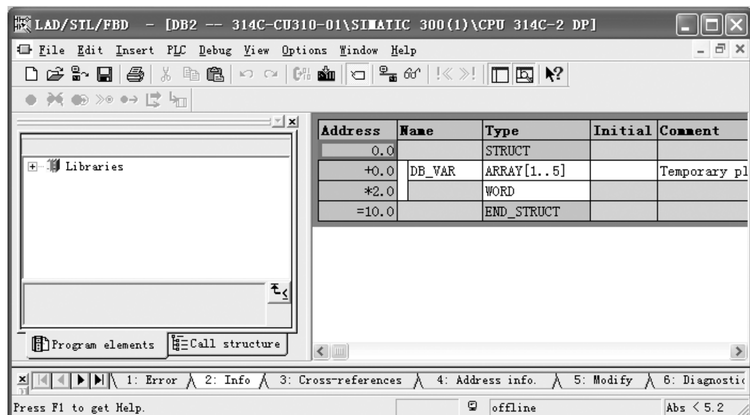



图 11-52　创建数组

③ 参数表赋值。打开参数表，输入如图 11-53 所示的参数和参数值。注意，“W#16#047E”的含义和“W#16#1000”的含义可参考 11.5.2，在此不作赘述。而 M1.0 是起动的开关，图中是“false”（假），当参数表处于监控状态时，将 M1.0 修改为“true”（真），则伺服电动机以一定的速度运行。

Var - [VAT_1 -- 314C-CU310-01\SIMATIC 300(1)\CPU 314C-2 DP\S7 ...

Table Edit Insert PLC Variable View Options Window Help

| | Address | Symbol | Display format | Status value | Modify value |
|---|---|---|---|---|---|
| 1 | DB2.DBW 0 | | HEX | W#16#0E7F | W#16#047E |
| 2 | DB2.DBW 2 | | HEX | W#16#1000 | W#16#1000 |
| 3 | M 1.0 | | BOOL | | true |
| 4 | DB1.DBW 0 | | HEX | | |
| 5 | DB1.DBW 2 | | HEX | | |
| 6 | | | | | |

Press F1 for help. Offline Abs < 5.2

图 11-53　参数表赋值

④ 编写程序。按照如图 11-54 编写程序，注意 M1.0 和 M1.1 是脉冲时才有效。

**程序段 1**：标题：

接收来自伺服驱动器的信息读入到DB1.DBB0开始的4个字节

SFC14
Read Consistent
Data of a Standard
DP Slave
"DPRD_DAT"
M1.1
EN　ENO
W#16#100 — LADDR　RET_VAL — MW12
RECORD — P#DB1.DBX0.0 BYTE 4

**程序段 2**：标题：

将DB1.DBB0开始的4个字节写入到伺服驱动器中

SFC15
Write Consistent
Data to a Standard
DP Slave
"DPWR_DAT"
M1.0
EN　ENO
W#16#100 — LADDR　RET_VAL — MW6
P#DB2.DBX0.0 BYTE 4 — RECORD

图 11-54　OB1 中的程序

（5）下载和调试

将编译后的硬件、程序、数据块和参数表全部下载到 CPU 314C－2DP 中，注意不能缺少其中任何一个。单击参数表中的“参数监控”按钮，使参数表处于监控状态，再将 M1.0 的参数 false 改为 true，最后单击“更新参数”按钮，更新参数，伺服电动机旋转。DB1 中显示伺服驱动反馈的参数，如图 11-55 所示。

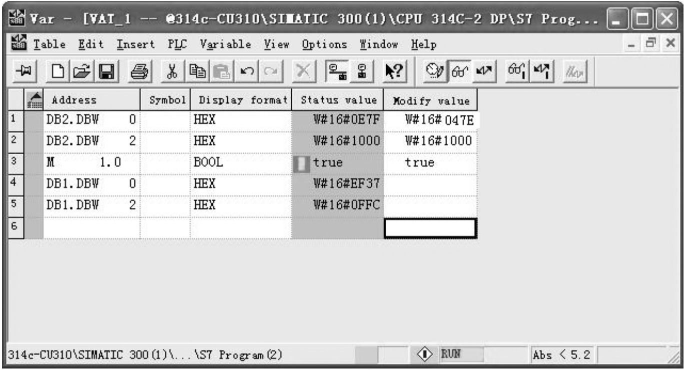



图 11-55　参数表处于监控状态

西门子的伺服系统功能强大，需要设置的参数较多，因此使用比较复杂，请读者参考相关说明书。若读者已经掌握了西门子变频器，无疑有利于读者掌握西门子伺服系统。

## 11.6　变频器的调试

MM440 变频器的标准供货方式装有状态显示板（SDP），状态显示板的内部没有任何电路，因此要对变频器进行调试，通常采用基本操作面板（BOP）、高级操作面板（AOP）和计算机（PC）等方法进行调试。基本操作面板（BOP）和高级操作面板（AOP）是可选件，需要单独订货。使用计算机调试时，计算机中需要安装“Drive Monitor”“Starter”或者“Scout”。“Scout”功能强大，包含“Starter”软件。

基本操作面板（BOP）调试变频器在前面已经讲解了，以下仅介绍“Starter”调试变频器。

### 11.6.1　STARTER 软件与传动装置常用通信连接方式

STARTER 软件与传动装置可以通过以下三种常用的通信方式建立连接。

**1. RS－232 串口通信（USS 通信协议）**

需要使用 PC to MM440 组件（订货号：SSE6400－1PC00－0AA0），组件安装在 BOP 板的插孔（使用 BOP 链路），使用计算机的 RS－232C 接口即可，如果笔记本电脑没有 RS－232C 接口，也可以在笔记本电脑上使用 USB 转换器。上位机软件使用“Drive Monitor”或者“Starter”。

**2. RS－485 串口通信（USS 通信协议）**

MM440 的 29(P＋)和 30(N－)控制接线端子是用于 RS－485 串行通信的通信口（COM

链路)。其连接示意图如图 11-56 所示。采用这种连接方式调试时，切不可将 29 号接线端子和 30 号接线端子接反，否则将产生烧毁接口的严重后果。上位机软件使用“Drive Monitor”或者“STARTER”。

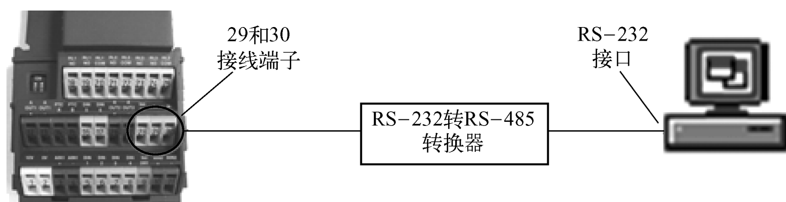



图 11-56　采用 RS-485 口进行 PC 调试连接图

**3. PROFIBUS 通信**

采用 PROFIBUS 通信协议调试 MM440 时，变频器上需要安装 PROFIBUS 通信模块（订货号：6SE6400-1PB00-0AA0)，计算机上需要安装 CP5611 或者 CP5621（此模块是目前新型号模块）等通信模块。其连接示意图如图 11-57 所示。CP5611（或 CP5621）与 PROFIBUS 通信模块的电缆就是购买 CP5611（或 CP5621）卡时配置的通信电缆，CP5611（或 CP5621）与变频器 PROFIBUS 通信模块均有 PROFIBUS 接口。

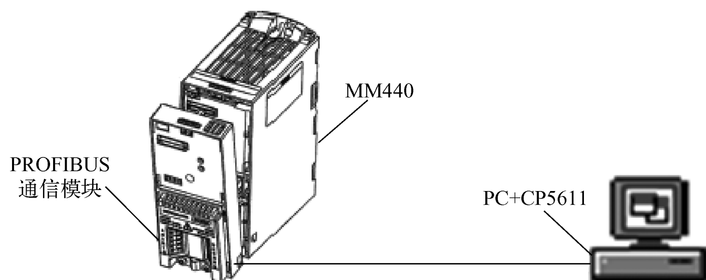



图 11-57　采用 PROFIBUS 口进行 PC 调试连接图

此外还可以用 USB 和以太网连接等形式进行调试，在此不作赘述。

## 11.6.2　用 STARTER 软件调试 MM440 实例

当控制系统使用的变频器数量较大且很多参数相同时，使用 PC 进行变频器调试，可以大大地节省调试时间，提高工作效率。以下通过例子介绍用 STARTER 软件调试 MM440 变频器。

**【例 11-11】**某设备上有一台 MM440，要求：对变频器进行参数设置。

**【解】**

**1. 软硬件配置**

① 1 套 STARTER 4.4（或者 SCOUT)。

② 1 台 MM440 变频器（含 PROFIBUS 卡)。

③ 1 块 CP5611/5621。

④ 1 台电动机。

在调试 MM440 变频器之前，将计算机和 MM440 变频器进行连接，如图 11-57 所示。

**2. 具体调试过程**

（1）新建项目

打开 STARTER 软件，新建项目，本例为“STARTER”存储在 D 盘。单击“Continue”（继续）按钮，如图 11-58 所示。

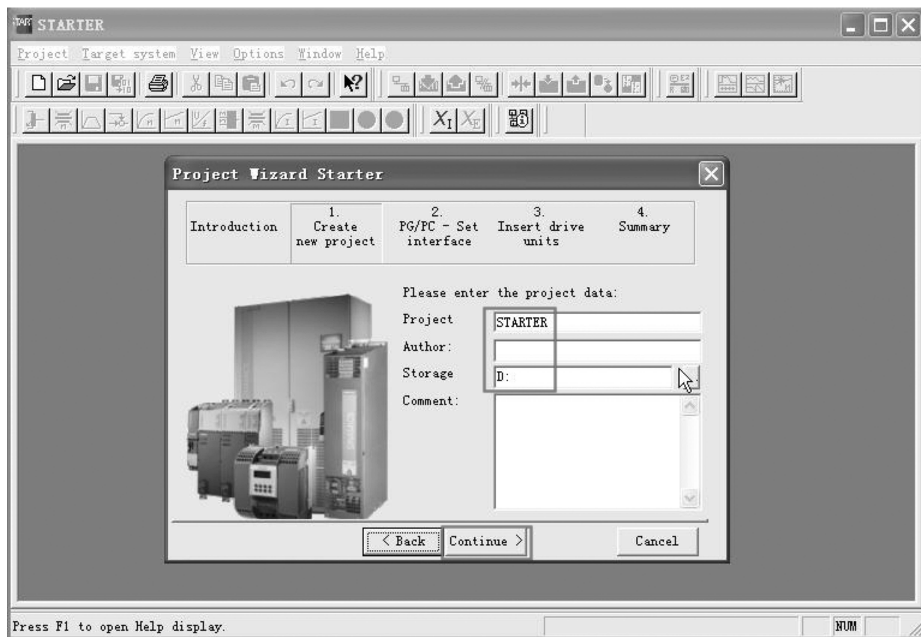



图 11-58　新建项目

（2）设置“PC/PG”接口

选中“CP5611（PROFIBUS）<Activate>”选项，单击“OK”（确定）按钮，如图 11-59 所示。再单击“Continue”（继续）按钮，如图 11-60 所示。

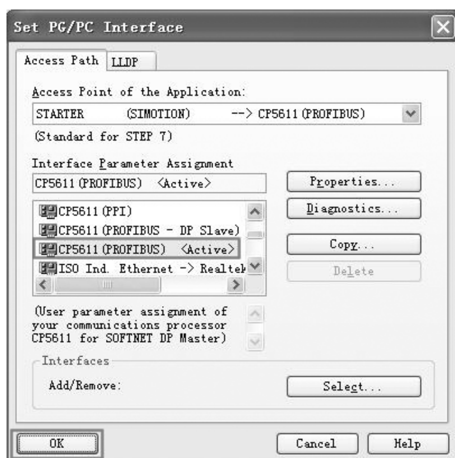



图 11-59　设置“PC/PG”接口（1）

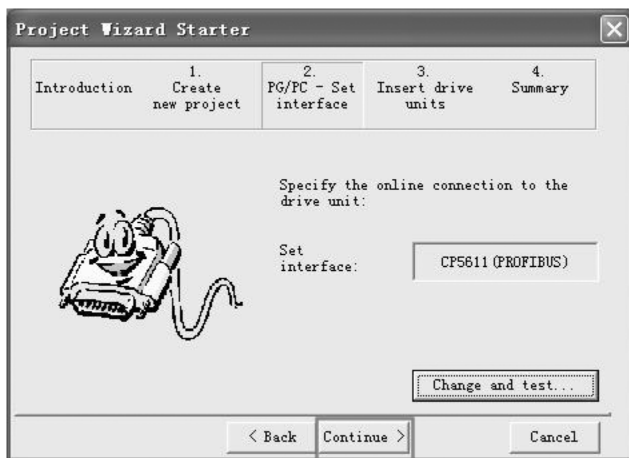



图 11-60　设置“PC/PG”接口（2）

（3）建立 PC 与 MM440 的在线连接。

STARTER 自动搜索 MM440，如图 11-61 所示，当找到 MM440（PC 与 MM440 已经建立连

接）时，跳出如图 11-62 所示的界面。单击“Complete”（完成）按钮，如图 11-63 所示。

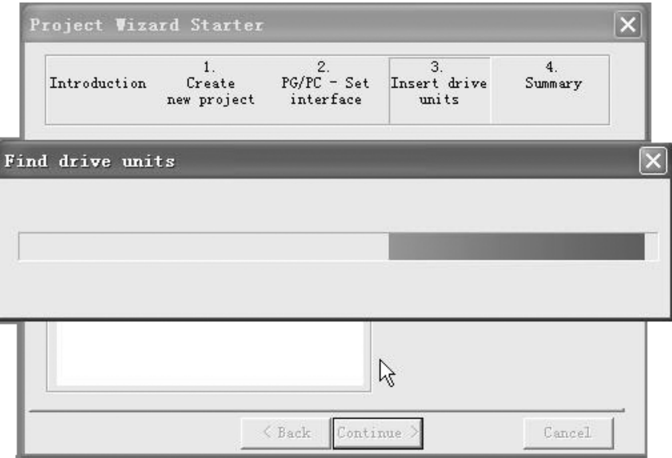



图 11-61　搜索 MM440

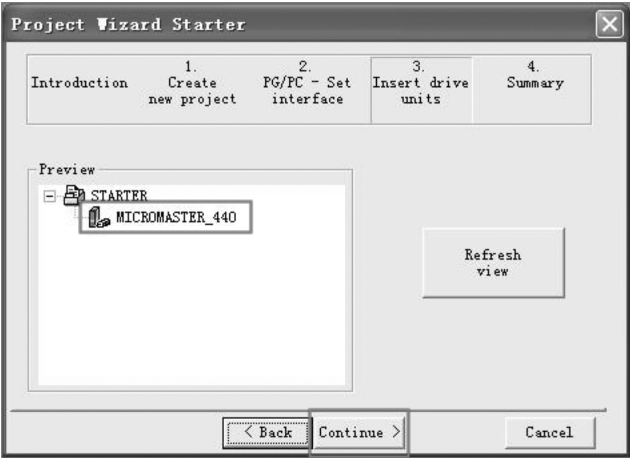



图 11-62　PC 与 MM440 已经建立连接（1）




图 11-63　PC 与 MM440 已经建立连接（2）

（4）变频器参数设置

单击“在线”按钮，使得变频器处于“在线”状态，在界面的最下方有“Online mode”（在线模式）显示，如图 11-64 所示。双击目录树“MICROMASTER 440”下层的“Configuration”（配置），如图 11-65 所示，开始配置参数。

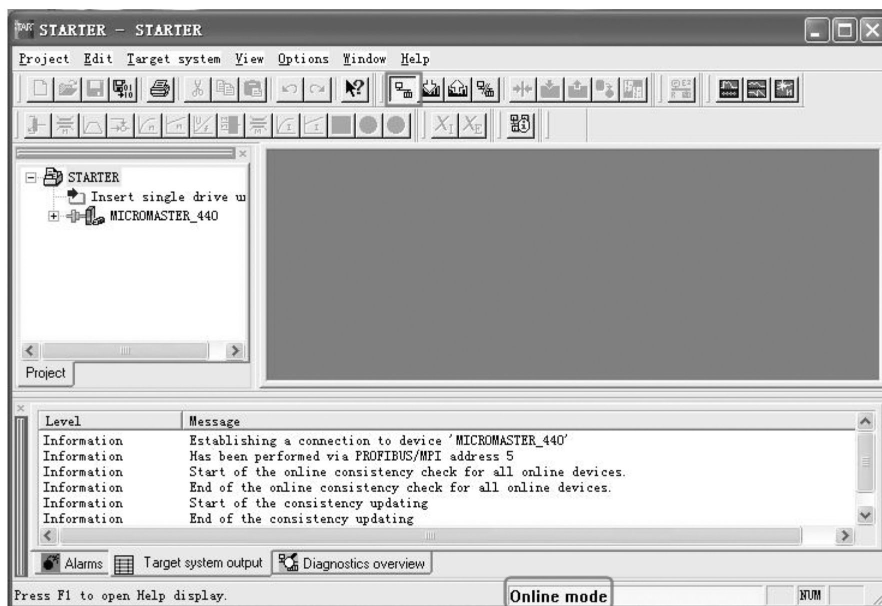



图 11-64　MM440 处于在线模式

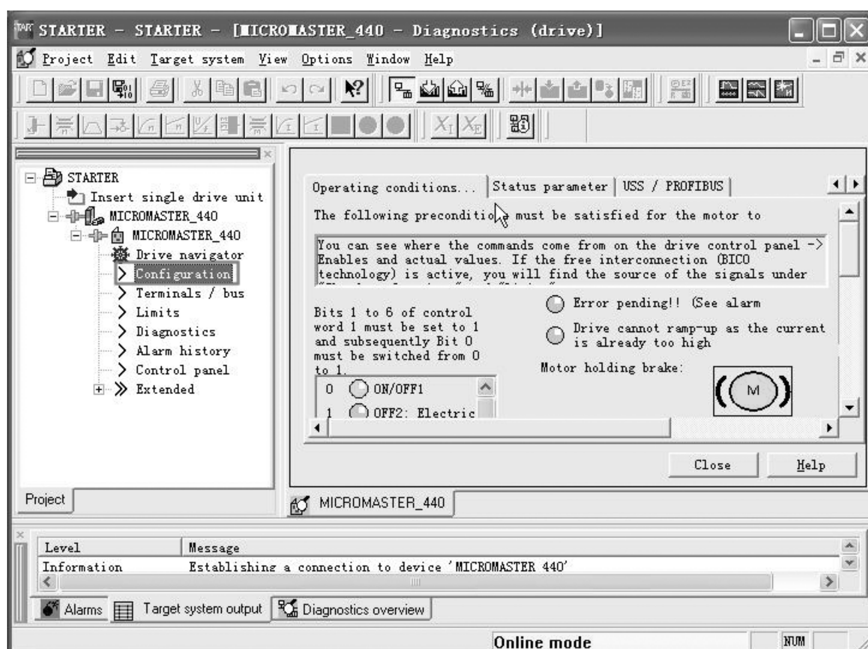



图 11-65　PC 与 MM440 已经建立连接

1）电网标准和负载类型选择。单击“Reconfigure drive”（重新配置驱动）按钮，如图 11-66 所示。如图 11-67 所示，在“1”处，选择工作地区电网标准，本例选择是欧洲标准。在“2”处，选择应用对象类型，即负载类型，本例选择“Constant torque”（恒扭矩）。

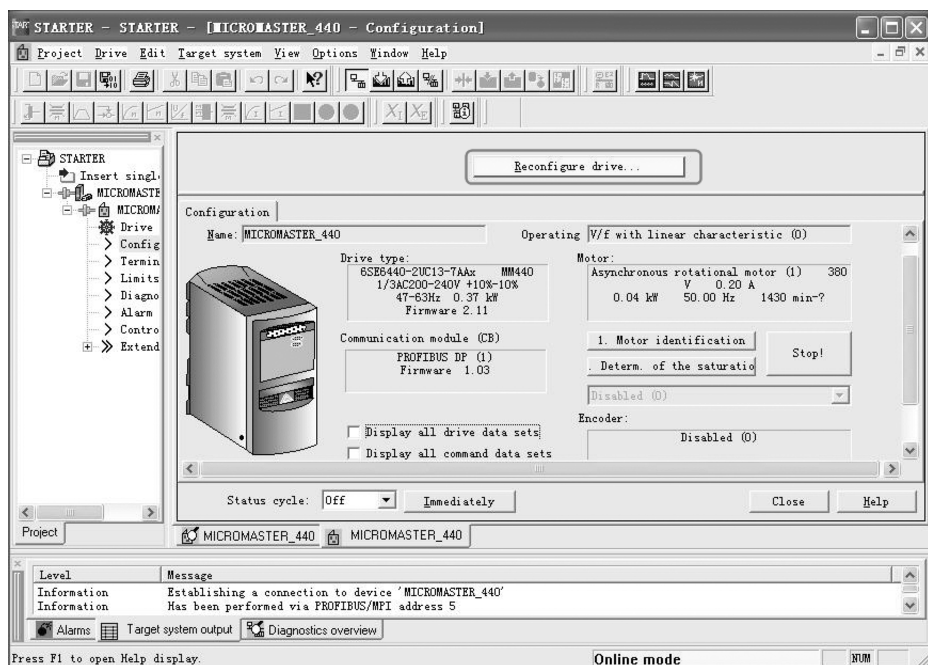



图 11-66　重新配置参数

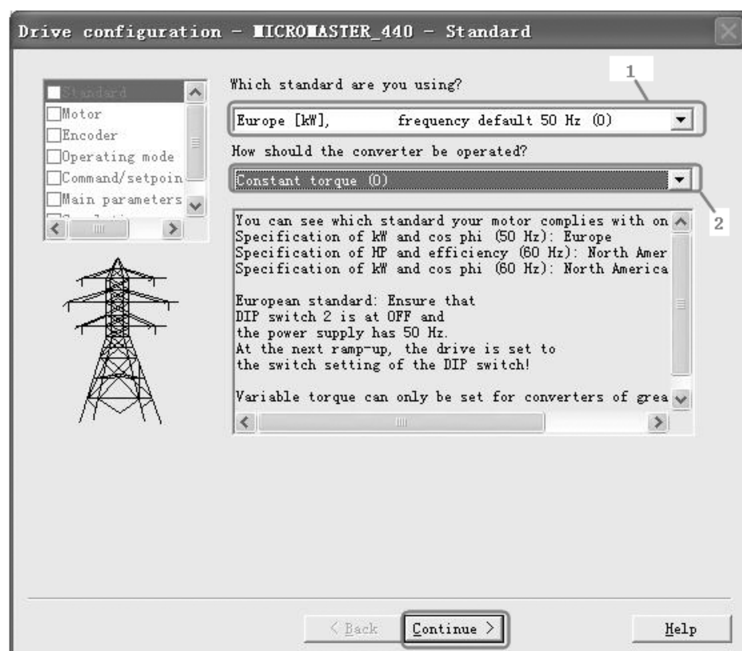



图 11-67　电网标准和负载类型选择

2）电动机参数的设置。设置电动机的类型（本例为异步电动机）、额定电压、额定电流、额定功率、额定转速、额定频率、功率因素和制冷方式（本例为自制冷），单击“Continue”（继续）按钮，如图 11-68 所示。

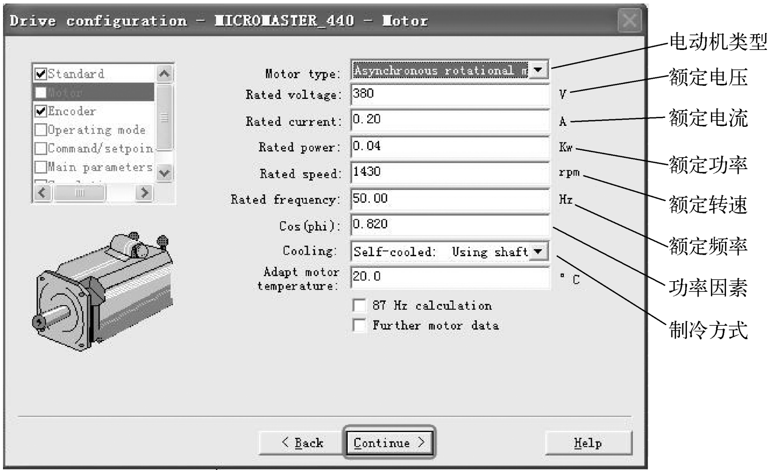



图 11-68　设置电动机参数

3）编码器设置。本例没有编码器，选择“Disabled”（封锁），单击“Continue”（继续）按钮，如图 11-69 所示。

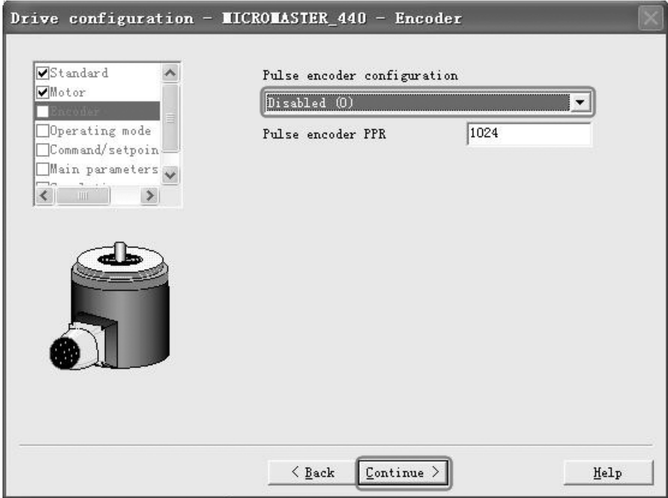



图 11-69　编码器设置

4）运行模式设置。选择“V/f”运行模式，如图 11-70 所示，单击“Continue”（继续）按钮。

5）设置主运行参数。如图 11-71 所示，选择“MOP Setpoint”，单击“Continue”（继续）按钮。在如图 11-72 所示的界面里，设置电动机过载系数、最小运行频率、最大运行频率、斜坡加速时间、斜坡减速时间和 OFF3 减速时间，单击“Continue”（继续）按钮。单击“Finish”（完成）按钮，保存系统参数设定，如图 11-73 所示。保存参数设置的运行过程，如图 11-74 所示。

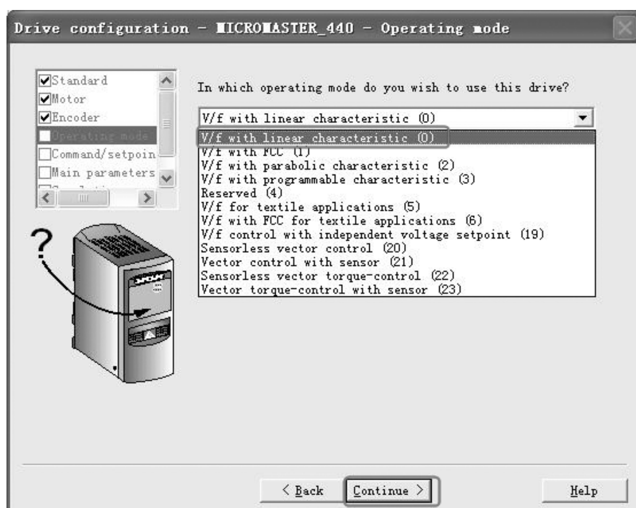



图 11-70 运行模式设置

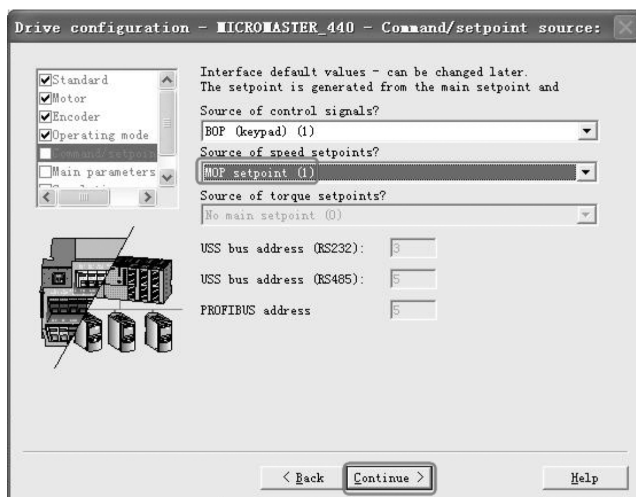



图 11-71 设置主运行参数（1）

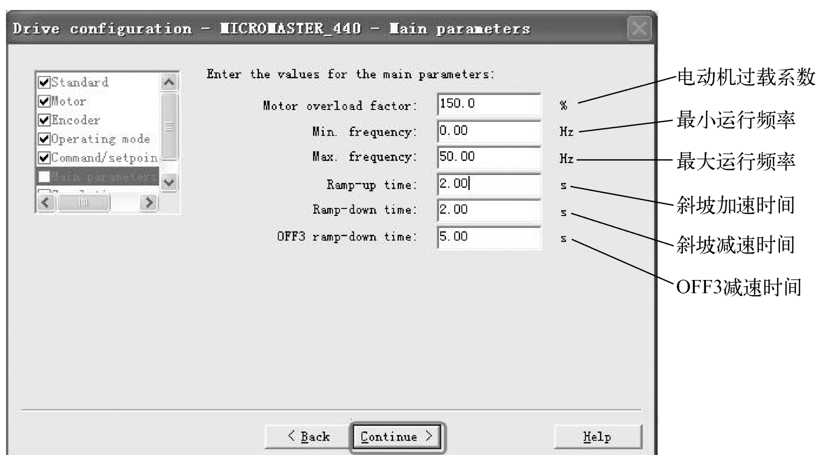



图 11-72 设置主运行参数（2）

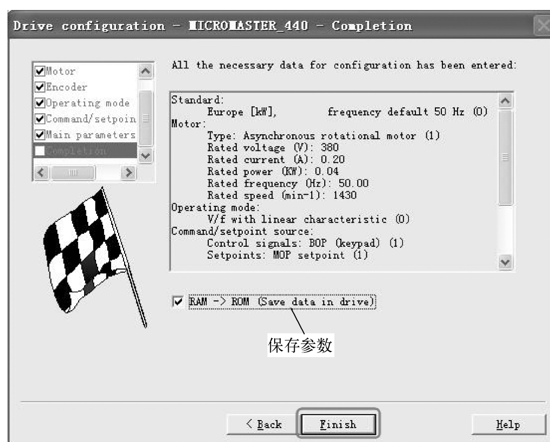



图 11-73　保存设置参数

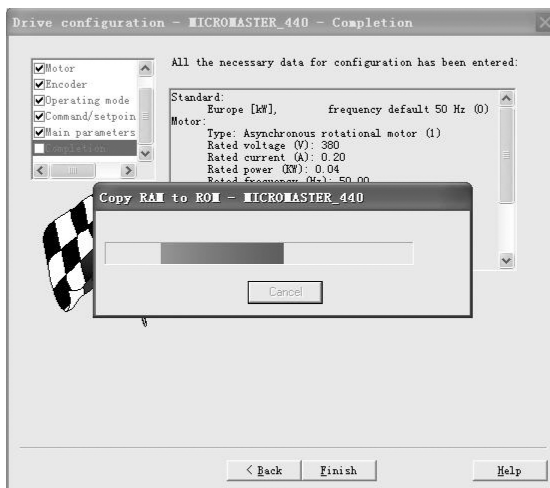



图 11-74　保存设置参数的过程

6）电动机的参数识别与优化。单击“Motor identification”，再单击“OK”（确定）按钮，如图 11-75 所示。

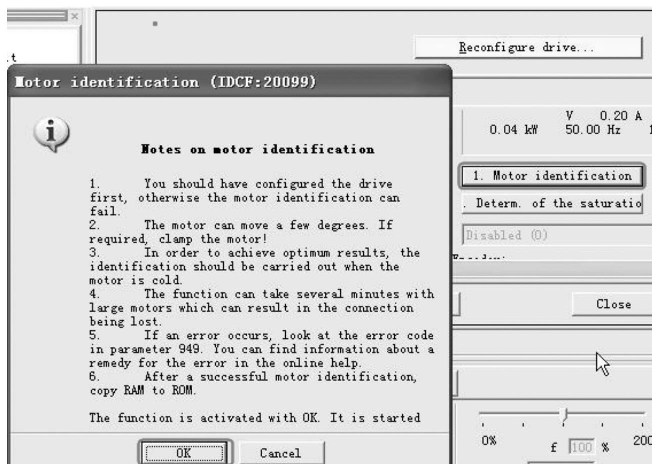



图 11-75　保存设置参数

7）使用 Control Panel 功能。双击“Control Panel”，如图 11-76 所示。单击“Accept”（接受）按钮，如图 11-77 所示。

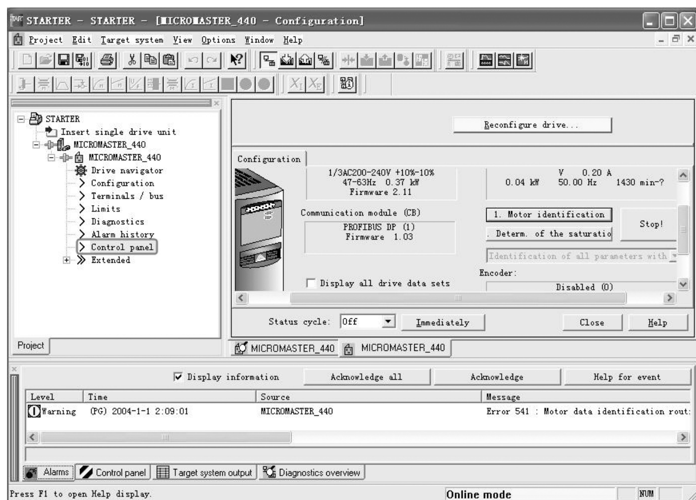



图 11-76　使用 Control Panel 功能（1）

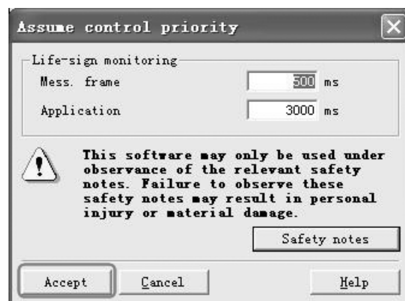



图 11-77　使用 Control Panel 功能（2）

勾选“Enable”选项，再设置运行频率，本例为 18 Hz，按下按钮“I”，电动机运行，按下按钮“O”，电动机停止运行，如图 11-78 所示。




图 11-78　使用 Control Panel 功能

8）上传备份变频器参数。单击“上传”按钮，如图 11-79 所示，单击“Yes”按钮。

当参数上传完成，单击“OK”按钮，如图 11-80 所示。




图 11-79　上传备份变频器参数




图 11-80　上传备份变频器参数完成

# 第 12 章　西门子 S7－300/400 PLC 的故障诊断技术

本章介绍 PLC 的故障诊断方法，重点介绍 S7－300/400 PLC 控制的故障诊断方法，最后介绍 30 个故障实例。

## 12.1　PLC 控制系统的故障诊断概述

PLC 是运行在工业环境中的控制器，一般而言可靠性比较高，尽管出现故障的概率较低，但出现故障也是难以避免的。引发故障的原因有很多，故障的后果也有很多种。

引发故障的原因虽然我们不能完全控制，但是可以通过日常的检查和定期的维护来消除多种隐患，把故障率降到最低。故障后果轻的可能造成设备停机，影响生产数量；重的可能造成财产损失和人员伤亡，如果是一些特殊的控制对象，一旦出现故障可能会引发更严重的后果。

故障发生后，对于维护人员最重要的是找到故障原因，迅速排除故障，尽快恢复系统的运行。对于系统设计人员在设计时要考虑到系统出现故障后的系统自我保护措施，力争使故障的停机时间最短，故障造成的损失最小。

一般 PLC 的故障主要由外部故障或内部错误造成。外部故障是由外部传感器或执行机构的故障等引发的，可能会使整个系统停机，甚至烧坏 PLC。而内部错误是 PLC 内部的功能性错误或编程错误造成的，可以使系统停机。S7－300/400 PLC 具有很强的错误（或称故障）检测和处理能力，CPU 检测到某种错误后，操作系统调用对应的组织块，用户可以在组织块中编程，对发生的错误采取相应措施。对于大多数错误，如果没有在组织块中编程，出现错误时 CPU 将进入 STOP 模式。

### 12.1.1　引发 PLC 故障的外部因素

**1. 外部电磁感应干扰**

PLC 外部存在干扰源，通过辐射或者电源线侵入 PLC 内部，引发 PLC 误动作或者造成 PLC 不能正常工作、停机，严重时，甚至烧毁 PLC。常见的措施如下：

1）PLC 周围有接触器等感性负载，可加冲击电压吸收装置，如 RC 灭弧器。

2）缩短输入和输出线的距离，并与动力线分开。

3）模拟量、通信线等信号应采用屏蔽线。线路较长时，可以采用中继方式。

4）PLC 的接地端子不能和动力线混用接地。

5）PLC 的输入端可以接入滤波器，避免从输入端引入干扰。

**2. 外部环境**

1）对于振动大的设备，安装电柜需要加橡胶垫等防振垫。

2）潮湿、腐蚀和多尘的场合容易造成生锈、接触不良、绝缘性能降低和短路等故障，这种情况应使用密封控制柜，有时还要采用户外型电器等特殊电器。

3）对于温度较高的场合，应加装排风扇，温度过高的场合则要加装空调。温度过低的场合还要加装加热器。

**3. 电源异常**

主要有缺相、电压波动或停电等，这些故障多由风、雪和雷电造成。常见措施如下：

1）直接起动电动机而造成的回路电压下降，PLC 回路应尽量与其分离。

2）PLC 的供电回路采用独立的供电回路。

3）选用 UPS 供电电源，提高供电可靠性和供电质量。

**4. 雷击、感应电**

雷击、感应电形成的冲击电压有时也会造成 PLC 损毁。

1）在 PLC 的输入端加压敏电阻等吸收元件。

2）加装浪涌吸收器或者氧化锌避雷器。

## 12.1.2 PLC 的故障类型和故障信息

**1. PLC 故障类型**

PLC 控制系统的硬件包括电源模块、I/O 模块、现场输入/输出元器件以及一些导线、接线端子和接线盒。

PLC 控制系统的故障是 PLC 故障和外围故障的总和。外围故障也会造成 PLC 的故障。PLC 控制系统的故障也可分为软件故障和硬件故障，其中硬件故障占 80%。

PLC 控制系统的故障的分布：

1）CPU 模块故障 5%。

2）单元故障 15%。

3）系统布线故障 5%。

4）输出设备故障 30%。

5）输入设备故障 45%。

控制系统故障 20% 由恶劣环境造成，80% 由用户使用不当造成。

**2. PLC 控制系统故障的分布与分层**

PLC 的外设故障占 95%，外设故障主要是继电器、接触器、接近开关、阀门、安全保护、接线盒、接线端子、螺纹连接、传感器、电源、电线和地线等。

PLC 自身故障占 5%，其中 90% 为 I/O 模块的故障，仅有 10% 是 CPU 模块的故障。首先将故障分为三个层次，第一层：是外部，还是内部故障；第二层：是 I/O 模块，还是控制器内部；第三层：是软件，还是硬件故障。

（1）第一层

利用 PLC 输入、输出 LED 灯判断是否为第一层故障。

（2）第二层

利用上位监控系统判断第二层次的故障，例如：M0.0 是输入，显示为 ON，Q0.0 显示为 ON，表示输入和输出都有信号，但 PLC 无输出，则判断 PLC 的外围有故障。

（3）第三层

例如清空 PLC 中的程序，下载一个最简单的程序到 PLC 中，如 PLC 能正常运行，则大致判断 PLC 正常。

**3. PLC 控制系统最易发生故障的部分**

（1）电源和通信系统

PLC 的电源是连续工作的，电压和电流的波动造成冲击不可避免，据 IBM 统计大约有 70% 以上的故障归根结底源自工作电源。

外部的干扰是造成通信故障的主要原因，此外经常插拔模块、印制电路板的老化和各种环境因素都会影响内部总线通信。

（2）PLC 的 I/O 端口

I/O 模块的损坏是 PLC 控制系统中较为常见的，减少 I/O 模块的损坏首先要正确设计外部电路，不可随意减少外部保护设备，其次对外部干扰因素进行有效隔离。

（3）现场设备

### 12.1.3 PLC 故障诊断方法

**1. PLC 故障的分析方法**

通常全局性的故障一般会在上位机上显示多处元件不正常，这通常是 CPU、存储器、通信模块和公共电源等发生故障。PLC 故障分析方法如下：

1）根据上位机的故障信息查找，准确而且及时。

2）根据动作顺序诊断故障，比较正常和不正常动作顺序，分析和发现可疑点。

3）根据 PLC 的输入/输出口状态诊断故障。如果是 PLC 自身故障，则不必查看程序即可查询到故障。

4）通过程序查找故障。

**2. 电源故障的分析方法**

PLC 的电源为 DC 24V，范围是 24 V ±5%，而是 AC 220V 范围是 220 V ±10%。

当主机接上电源，指示灯不亮，可能的原因有：若拔出 +24 V 端子，指示灯亮，表明 DC 负载过大，这种情况，不要使用内部 24 V 电源；若拔出 +24 V 端子，指示灯不亮，则可能熔体已经烧毁，或者内部有断开的地方。

当主机接上电源，指示灯 POWER 闪亮，则 +24 V 和 COM 短路了。

BATT 灯亮表明锂电池寿命结束，要尽快更换电池。

**3. PLC 电源的抗干扰**

PLC 电源的抗干扰处理的方法如下：

1）控制器、I/O 电源和其他设备电源分别用不同的隔离变压器供电会更好。

2）控制器的 CPU 用一个开关电源，外部负载用一个开关电源。

PLC 电源的抗干扰处理后的典型例子如图 12-1 所示。

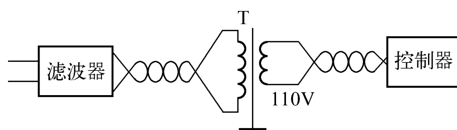



图 12-1 PLC 电源的抗干扰

### 12.1.4 PLC 外部故障诊断方法

**1. 输入给 PLC 信号出错的原因**

1）信号线的短路或者断路，主要原因是老化、拉扯、压砸线路及震动。

2）机械触头抖动。机械抖动压下一次，PLC 可能认为抖动了几次，硬件虽然加了滤波或者软件增加了微分，但由于 PLC 扫描周期短，仍然会影响计数、移位等。

3）现场传感器、继电器等损坏。

**2. 执行机构出错的可能原因**

1）输出负载没有可靠工作，如 PLC 已经发出信号，但继电器没有工作。

2）PLC 自身故障，从而负载不动作。

3）电动阀该动作没动作，或者没到位。

**3. PLC 控制系统布线抗干扰措施**

（1）电源的接线和接地

1）电源隔离器两端尽量采用双绞线，或者屏蔽电缆；电源线和 I/O 线要尽量分开布置。

2）交流和直流线要分别使用不同的电缆，分开捆扎、分槽走线。

3）共同接地是传播干扰的常见措施。应将动力线的接地和控制接地分开，动力线的接地应接在地线上，PLC 的接地接在机柜壳体上。要保证 PLC 控制系统的接地线和动力线的屏蔽线尽量等电位。

（2）输入和输出布线

PLC 的输入线指外部传感器、按钮等与 PLC 的输入接口的接线。开关量信号一般采用普通电缆，如距离较远则要采用屏蔽电缆。高速信号和模拟量信号应采用屏蔽电缆。不同的信号线最好不要公用同一接插件，以减少相互干扰。

尽量减少配线回路的距离。输入和输出信号电缆穿入专用的电缆管或者独立的线槽中敷设。当信号距离较远时如 300 m，可以采用中间继电器转接信号。通常布线要注意以下几点：

1）输入线的长度一般不长于 30 m。良好的工作环境下，距离可以适当加长。

2）输入线和输出线不能使用同一电缆，应分开走线，开关量和模拟量要分开敷设。

3）输入线和输出回路配线时，多股线必须压接接线端子或者单股线，多股线的直接与 PLC 端子压接时，容易产生火花。

**4. 外部故障的排除方法详细说明和处理**

PLC 有很强的自诊断能力，当 PLC 自身故障或外围设备发生故障，都可用 PLC 上具有诊断指示功能的发光二极管的亮灭来诊断。

（1）故障查找

根据总体检查流程图找出故障点的大方向，逐渐细化，以找出具体故障，如图 12-2 所示。

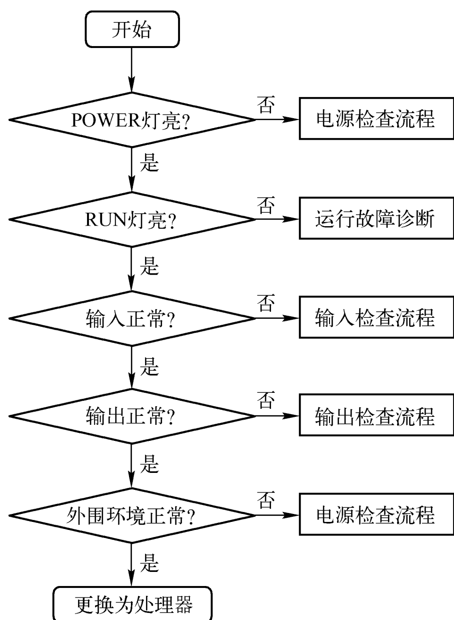


图 12-2　总体检查流程图

（2）故障的处理

不同故障产生的原因不同，对此有不同的处理方法，CPU 装置、I/O 扩展装置故障处理见表 12-1。

**表 12-1　CPU 装置、I/O 扩展装置故障处理**

| 序号 | 异常现象 | 可能原因 | 处　　理 |
|---|---|---|---|
| 1 | ［POWER］LED 灯不亮 | 1. 电压切换端子设定不良<br>2. 熔丝熔断 | 正确设定切换端子<br>更换熔丝 |
| 2 | 熔丝多次熔断 | 1. 电压切换端子设定不良<br>2. 线路短路或烧坏 | 正确设定切换端子<br>更换电源单元 |
| 3 | ［RUN］LED 灯不亮 | 1. 程序错误<br>2. 电源线路不良<br>3. I/O 单元号重复<br>4. 远程 I/O 电源关，无终端 | 修改程序<br>更换 CPU 单元<br>修改 I/O 单元号<br>接通电源 |
| 4 | 运行中输出端没闭合（［POWER］灯亮） | 电源回路不良 | 更换 CPU 单元 |
| 5 | 编号以后的继电器不动作 | I/O 总线不良 | 更换基板单元 |
| 6 | 特定的继电器编号的输出（入）接通 | I/O 总线不良 | 更换基板单元 |
| 7 | 特定单元的所有继电器不接通 | I/O 总线不良 | 更换基板单元 |

输入单元故障的处理见表 12-2。

**表 12-2　输入单元故障处理**

| 序号 | 异常现象 | 可能原因 | 处　　理 |
|---|---|---|---|
| 1 | 输入全部不接通（动作指示灯也灭） | 1. 未加外部输入电压 | 供电 |
|  |  | 2. 外部输入电压低 | 加额定电源电压 |
|  |  | 3. 端子螺钉松动 | 拧紧 |
|  |  | 4. 端子板联接器接触不良 | 把端子板重新插入、锁紧。更换端子板联接器 |
| 2 | 输入全部断开（输入指示灯也灭） | 输入回路不良 | 更换单元 |
| 3 | 输入全部不关断 | 输入回路不良 | 更换单元 |
| 4 | 特定继电器编号的输入不接通 | 1. 输入器件不良 | 更换输入器件 |
|  |  | 2. 输入配线断线 | 检查输入配线 |
|  |  | 3. 端子螺钉松动 | 拧紧 |
|  |  | 4. 端子板联接器接触不良 | 把端子板补充插入、锁紧。更换端子板联接器 |
|  |  | 5. 外部输入接触时间短 | 调整输入组件 |
|  |  | 6. 输入回路不良 | 更换单元 |
|  |  | 7. 程序的 OUT 指令中用了输入继电器编号 | 修改程序 |
| 5 | 特定继电器编号的输入不关断 | 1. 输入回路不良 | 更换组件 |
|  |  | 2. 程序的 OUT 指令中用了输入继电器编号 | 修改程序 |
| 6 | 输入不规则 ON/OFF 动作 | 1. 外部输入电压低 | 使外部输入电压在额定值范围 |
|  |  | 2. 噪声引起的误动作 | 抗干扰措施：<br>安装绝缘变压器、安装尖峰抑制器、用屏蔽线配线等 |
|  |  | 3. 端子螺钉松动 | 拧紧 |
|  |  | 4. 端子板联接器接触不良 | 把端子板补充插入、锁紧。更换端子板联接器 |

（续）

| 序号 | 异常现象 | 可能原因 | 处理 |
|---|---|---|---|
| 7 | 异常动作的继电器编号为 8 点单位 | 1. COM 端螺钉松动 | 拧紧 |
| | | 2. 端子板联接器接触不良 | 把端子板补充插入、锁紧。更换端子板联接器 |
| | | 3. CPU 不良 | 更换 CPU 单元 |
| 8 | 输入动作指示灯不亮（动作正常） | LED 灯坏 | 更换单元 |

输出单元故障的处理见表 12-3。

**表 12-3 输出单元故障处理**

| 序号 | 异常现象 | 可能原因 | 处理 |
|---|---|---|---|
| 1 | 输出全部不接通 | 1. 未加负载电源 | 加电源 |
| | | 2. 负载电源电压低 | 使电源电压为额定值 |
| | | 3. 端子螺钉松动 | 拧紧 |
| | | 4. 端子板联接器接触不良 | 把端子板补充插入、锁紧。更换端子板联接器 |
| | | 5. 熔丝熔断 | 更换熔丝 |
| | | 6. I/O 总线接触不良 | 更换单元 |
| | | 7. 输出回路不良 | 更换单元 |
| | | 8. 输出点烧毁 | 更换输出模块 |
| 2 | 输出全部不关断 | 输出回路不良 | 更换单元 |
| 3 | 特定继电器编号的输出不接通（动作指示灯灭） | 1. 输出接通时间短 | 更换单元 |
| | | 2. 程序中指令的继电器编号重复 | 修改程序 |
| | | 3. 输出回路不良 | 更换单元 |
| 4 | 特定继电器编号的输出不接通（动作指示灯亮） | 1. 输出器件不良 | 更换输出器件 |
| | | 2. 输出配线断线 | 检查输出线 |
| | | 3. 端子螺钉松动 | 拧紧 |
| | | 4. 端子联接接触不良 | 端子充分插入、拧紧 |
| | | 5. 继电器输出不良 | 更换继电器 |
| | | 6. 输出回路不良 | 更换单元 |
| 5 | 特定继电器编号的输出不关断（动作指示灯灭） | 1. 输出继电器不良 | 更换继电器 |
| | | 2. 由于漏电流或残余电压而不能关断 | 更换负载或加假负载电阻 |
| 6 | 特定继电器编号的输出不关断（动作指示灯亮） | 1. 程序 OUT 指令的继电器编号重复 | 修改程序 |
| | | 2. 输出回路不良 | 更换单元 |
| 7 | 输出出现不规则的 ON/OFF 现象 | 1. 电源电压低 | 调整电压 |
| | | 2. 程序 OUT 指令的继电器编号重复 | 修改程序 |
| | | 3. 噪声引起的误动作 | 抗噪声措施：装抑制器、装绝缘变压器、用屏蔽线配线等 |
| | | 4. 端子螺钉松动 | 拧紧 |
| | | 5. 端子联接接触不良 | 端子充分插入、拧紧 |

（续）

| 序号 | 异常现象 | 可能原因 | 处理 |
|---|---|---|---|
| 8 | 异常动作的继电器编号为8点单位 | 1. COM端子螺钉松动 | 拧紧 |
| | | 2. 端子联接接触不良 | 端子充分插入、拧紧 |
| | | 3. 熔丝熔断 | 更换熔丝 |
| | | 4. CPU不良 | 更换CPU单元 |
| 9 | 输出指示灯不亮（动作正常） | LED灯坏 | 更换单元 |

## 12.2　S7－300/400故障诊断技术

S7－300/400 PLC有强大的故障诊功能，除了12.1节中列举的故障诊断和处理方法外，还有自身的特色，下面将详细讲解。

### 12.2.1　使用状态和出错LED进行故障诊断

可以利用CPU面板上的指示灯进行初步诊断，同时可以使用STEP 7软件的诊断功能进行诊断，快速查找故障原因。

这种诊断方法简单、方便、直观，但是某些LED给出的故障信号可能很笼统，需要进一步使用其他诊断方法，例如用STEP 7的诊断视图和模块信息进行诊断，才能获得具体、准确的诊断信息。如果控制系统的分布范围很宽，查看所有设备的LED是很费时费事的。

**1. 使用S7－300 CPU状态和出错LED进行故障诊断**

S7－300 CPU的指示灯与CPU状态的关系见表12-4。

**表12-4　S7－300 CPU的指示灯与CPU状态关系**

| LED状态（空白表示与此无关） | | | | | 含义 |
|---|---|---|---|---|---|
| SF | 5VDC | FRCE | RUN | STOP | |
| 关 | 关 | 关 | 关 | 关 | CPU无电源 |
| 关 | 开 | | 关 | 开 | CPU处于STOP模式，正常状态 |
| 开 | 开 | | 关 | 开 | CPU因错处于STOP模式 |
| | 开 | | 关 | 闪烁（0.5 Hz） | CPU请求存储器复位 |
| | 开 | | 关 | 闪烁（2 Hz） | CPU正在执行存储器复位 |
| | 开 | | 闪烁（2 Hz） | 开 | CPU正在处于起动状态 |
| | 开 | | 闪烁（0.5 Hz） | 开 | CPU被编程设备的断电命令暂停 |
| 开 | 开 | | | | 硬件或者软件错误 |
| | | 开 | | | 启用“强制”功能 |
| | | 闪烁（2 Hz） | | | 激活了节点闪烁测试 |
| 闪烁 | 闪烁 | 闪烁 | 闪烁 | 闪烁 | CPU |

（1）SF 红色时可能的软件错误

1）启用和触发了日期中断，但未装载日期中断组织块 OB10。

2）用了 SFC32 触发延时中断，但未装载延时中断组织块 OB20。

3）触发了硬件中断，但未装载组织块 OB40。

4）调用了太多的 OB 块，超出了循环时间（默认为 150 ms）。

5）编程错误。未加载块，定时器等的编号错误，寄存器（如 M）的范围超出规定范围。

6）I/O 访问错误。

（2）SF 红色时可能的硬件错误

1）系统正常运行时，卸下或者插入模块（S7 -300 不支持热插拔）。

2）系统正常运行时，在 DP 总线上接入或者取下分布式模块。

3）系统正常运行时，在 PROFINET 总线上接入或者取下分布式模块。

4）MMC 卡故障。

5）模块松动时，导致不能识别硬件。

（3）BF 常亮时的故障

1）总线故障。

2）DP 接口故障。

3）多个站点有不同的波特率。

4）总线短路。

（4）BF 闪亮时的故障及其解决方案

1）主站可能的错误：组态错误、连接站有故障或者无法访问至少一个从站。要检查从站是否连接到主站，总线是否有断开。

2）从站可能的错误：总线连接器是否连接正确，总线是否断路。

**2. 使用 S7 -400 状态和出错 LED 进行故障诊断**

S7 -400 的指示灯与 CPU 状态的关系见表 12-5。

**表 12-5　S7 -300 的指示灯与 CPU 状态关系**

| 序号 | 指示灯 | 颜色 | 说　　明 |
|---|---|---|---|
| 1 | INTF | 红色 | 内部故障，例如程序运行超时 |
| 2 | EXTF | 红色 | 外部故障，例如电源或者模块故障 |
| 3 | FRCE | 黄色 | 输入、输出有强制 |
| 4 | RUN | 绿色 | 运行模式 |
| 5 | STOP | 黄色 | 停止模式 |
| 6 | CRST | 黄色 | 完全复位 |
| 7 | BUS1F | 红色 | 1 号通信口（MPI/DP）的总线故障 |
| 8 | BUS2F | 红色 | 2 号通信口（DP）的总线故障 |
| 9 | BUS5F | 红色 | PROFINET 接口的总线故障，组态了 PROFINET IO 系统但未进行连接 |
| 10 | IFM1F | 红色 | 存储器子模块 1 故障 |
| 11 | IFM2F | 红色 | 存储器子模块 2 故障 |

（1）INTF 红色可能的故障

1）超时错误（超过最大的循环时间）。出错时一般调用 OB80，程序中如没有下载 OB80，则 CPU 置于“STOP”状态。

2）CPU 硬件错误。如 MPI 接口故障、分布式 IO 故障时，或者故障消失时，调用 OB84 模块，程序中如没有下载 OB84，则 CPU 置于“STOP”状态。

3）通信错误。当接收全局数据时，检测到错误的标识符（ID）、帧长度错误、数据块不存在等，调用 OB87，程序中如没有下载 OB87，则 CPU 置于“STOP”状态。

4）有编程错误时，调用 OB121，程序中如没有下载 OB121，则 CPU 置于“STOP”状态。

5）块的调用嵌套深度过大，调用 OB88，程序中如没有下载 OB88，则 CPU 置于“STOP”状态。

（2）EXTF 红色可能的故障

1）I/O 访问错误，可能是模块故障，I/O 模块松动造成接触不良，或者 CPU 不能识别 I/O 地址。发生 I/O 访问错误时，调用 OB122。

2）电源故障的出现或者消失。备用电池失效或者未安装 24 V 电源故障。发生电源故障时，调用 OB81。

3）插入或者拔出模块。S7－400 允许带电热插拔，但要调用 OB83。

4）优先级错误。发生优先级错误时，调用 OB85。

5）机架/站故障。扩展机架故障或者远程 IO 故障。调用 OB86。

**3. 用 DP 从站的 LED（以 IM153－2 为例）进行故障诊断**

DP 从站的 LED（IM153－2）多用于工程中。DP 从站的指示灯及其含义见表 12-6。

**表 12-6　IM153－2 的指示灯含义**

| 序号 | 指示灯 | 颜色 | 说　明 |
|---|---|---|---|
| 1 | ON | 绿色 | 供电正常 |
| 2 | SF | 红色 | 组织错误 |
| 3 | BF | 红色 | DP 故障 |
| 4 | ACT | 黄色 | 冗余模式中的主动模式 |

各指示灯状态的含义见表 12-7。

**表 12-7　IM153－2 的指示灯状态组合含义**

| SF | BF | ACT | ON | 含　义 | 措　施 |
|---|---|---|---|---|---|
| 灭 | 灭 | 灭 | 灭 | IM153－2 没有通电，或者模块故障 | 接通电源或者更换模块 |
| 无关 | 无关 | 无关 | 亮 | IM153－2 通电，运行状态 | |
| 亮 | 灭 | 灭 | 灭 | 模块通电后，正在复位 | |
| 亮 | 亮 | 亮 | 亮 | 通电后正在硬件测试 | |
| 亮 | 0.5 Hz 闪亮 | 灭 | 灭 | 外部故障，使用了不适合操作系统的 MMC 卡 | 换 MMC 卡 |
| 亮 | 0.5 Hz 闪亮 | 灭 | 灭 | 内部故障，写入更新文件时的错误 | 重新更新。可能存储器损坏 |

（续）

| SF | BF | ACT | ON | 含　义 | 措　施 |
|---|---|---|---|---|---|
| 无关 | 闪烁 | 灭 | 亮 | 模块为正确组态，主站和从站之间没有数据交换，可能地址不正确，总线错误 | 检查组态和参数，电缆长度，中断电阻，波特率等 |
| 无关 | 亮 | 灭 | 亮 | DP 主站和模块无连接。总线通信已中断 | 检查总线连接器是否安装正确，电缆是否连接断开。断开 24 V 电源，重新接通 |
| 亮 | 闪烁 | 灭 | 亮 | 组态的 ET200 与实际的 ET200 不一致 | 更改组态，重新下载 |
| 亮 | 灭 | 灭 | 亮 | 无效的 DP 地址 | 检查 DP 地址 |
| 无关 | 灭 | 亮 | 亮 | IM153 -2 正在与 DP 主站和 ET200 的 I/O 交换数据 | |
| 无关 | 灭 | 灭 | 亮 | 电压已经供给 IM153 -2 | |
| 0.5 Hz 闪亮 | 灭 | 灭 | 亮 | 在冗余模式，IM153 -2 是被动模块 | |
| 闪烁 | 闪烁 | 闪烁 | 闪烁 | 当前运行模式的 IM153 -2 与冗余 IM153 -2 不兼容 | |

## 12.2.2　用 STEP 7 诊断视图进行故障诊断

S7 -300/400 PLC 具有非常强大的故障诊断功能，通过 STEP 7 编程软件可以获得大量的硬件故障与编程错误的信息，使用户能迅速地查找到故障所在。

这里的诊断是指 S7 -300 内部集成的错误识别和记录功能，错误信息在 CPU 的诊断缓冲区内。有错误或事件发生时，标有日期和时间的信息被保存到诊断缓冲区，时间保存到系统的状态表中，如果用户已对有关的错误处理组织块编程，CPU 将调用该组织块。

**1. 故障诊断的基本方法**

在 SIMATIC 管理器中用菜单命令“查看（View）”→“在线（Online）”打开在线窗口。打开所有的站，查看是否有 CPU 显示了指示错误或故障的诊断符号。

诊断符号用来形象直观地表示模块的运行模式和模块的故障状态，如图 12-3 所示。如果模块有诊断信息，在模块符号上将会增加一个诊断符号，或者模块符号的对比度降低。




图 12-3　诊断符号

1）诊断符号“当前组态与实际组态不匹配”，表示被组态的模块不存在，或者插入了与组态的模块的型号不同的模块。

2）诊断符号“故障”，表示模块出现故障。可能的原因为：诊断中断、I/O 访问错误或检测到错误 LED。

3）诊断符号“停止”，表示 CPU 处于停机状态。

4）诊断符号“运行”，表示 CPU 处于正常运行状态。

5）诊断符号“强制”，表示在该模块上有变量被强制，即在模块的用户程序中有变量被赋予一个固定植，该数据值不能被程序改变。“强制”符号可以与其他符号组合在一起显示。

6）诊断符号“保持”，该符号在进行断点调试时出现。

7）诊断符号“无法诊断”，表示无线上连接，或该模块不支持模块诊断信息，例如电源模块或子模块。

从在线的SIMATIC管理器的窗口、在线的硬件诊断功能打开的快速窗口和在线的硬件组态窗口（诊断窗口），都可以观察到诊断符号。

通过观察诊断符号，可以判断CPU模块的运行模式，是否有强制变量，CPU模块和功能模块（FM）受否有故障。

打开在线窗口，在SIMATIC管理器中执行菜单命令“PLC”→“诊断/设置”→“硬件诊断”，将打开硬件诊断快速浏览窗口。在该窗口中显示PLC的状态，看到诊断功能的模块的硬件故障，双击故障模块可以获得详细的故障信息。

**2. 利用CPU诊断缓冲区进行详细故障诊断**

建立与PLC的在线连接后，在SIMATIC管理器中选择要检查的站，执行菜单命令“PLC”→“诊断/设置”→“模块 信息”，如图12-4所示，打开模块信息窗口，显示该站中CPU的信息。在快速窗口中使用“模块信息”。



图12-4　打开CPU诊断缓冲区

在模块信息窗口中的诊断缓冲区（Diagnostic Buffer）选项中，给出了CPU中发生的事件一览表，选中“事件”窗口中某一行的某一事件，下面灰色的“关于事件的详细资料”窗口将显示所选事件的详细信息，如图12-5所示。使用诊断缓冲区可以对系统错误进行分析，查找停机的原因，并对出现的诊断时间分类。

**3. 用STEP 7诊断视图进行故障诊断的举例**

（1）存储卡故障

在SIMATIC Manager界面，单击“PLC”→“诊断/设置”→“硬件诊断”，如图12-6所示。

在硬件诊断界面，单击“模块信息”按钮，弹出如图12-7所示界面，显示模块是正常的。

在“诊断缓冲区”选项卡中，可以看到是编程错误，经过仔细核对编程正确，而实际是存储卡故障，如图12-8所示。换新MMC卡后，重新下载程序后正常。如果MMC卡没有

插好，也会显示此故障。




图 12–5 CPU 模块的在线模块信息窗




图 12–6 硬件诊断 – 快速查看




图 12–7 模块信息 – 常规




图 12-8　模块信息 - 诊断缓冲区

（2）未装载 FC

在“诊断缓冲区”选项卡中，可以看到是编程错误，存储卡正常，缓冲区明确指出未装载 FC，如图 12-9 所示。



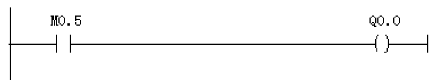

图 12-9　模块信息 - 诊断缓冲区

经过仔细核对，操作者编写的梯形图程序（如图 12-10 所示），的确未下载功能 FC1，但在主程序中又调用了 FC1，因而出错。

（3）硬件组态和实际配置的不匹配

在 SIMATIC Manager 界面中，单击“PLC”→“诊断/设置”→“硬件诊断”。可以看到 SM 模块上有个红色斜线，如图 12-11 所示。选中“SM 模拟量”模块，单击“模块信息”

按钮，弹出如图 12-12 的界面。注意：有时查看出错信息时，错误类型是“硬件组态和实际配置的不匹配”，但显示的不是红色斜杠，而是红色的叉号（×）。



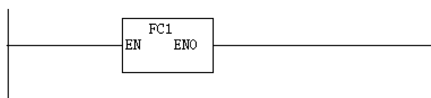




图 12-10 梯形图




图 12-11 硬件诊断 - 快速查看

在 12-12 中“状态”信息中显示：插入的模块与实际的模块型号不同，也就是硬件组态和实际配置的不匹配。

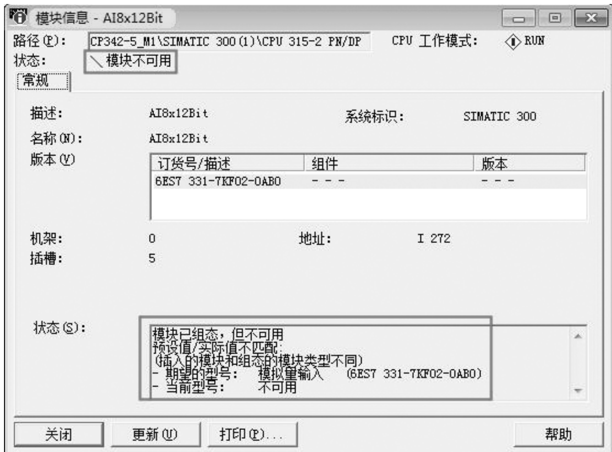



图 12-12 模块信息 - DP 从站

（4）I/O 访问出错

在 SIMATIC Manager 界面，单击“PLC”→“诊断/设置”→“硬件诊断”。可以看到 CPU 模块处于“STOP”状态，SM 和 CP 模块出错，如图 12-13 所示。选中“SM 模拟量”模块，单击“模块信息”按钮，弹出如图 12-14 所示的界面。

如图 12-14 所示，可以看到 I/O 访问故障，也就是 CPU 模块不能访问 SM 和 CP 模块的数据。I/O 访问故障的原因可能有：

1）模块损坏，或者模块中的插针歪斜。

2）模块松动。

3）U 型连接器损坏。

4）组态模块与实际安装模块不匹配等。

经检查发现本例组态模块的顺序为 CPU、SM、CP，而实际安装顺序 CPU、CP、SM，因此造成 I/O 访问故障，可见模块出错有时并非单一原因。

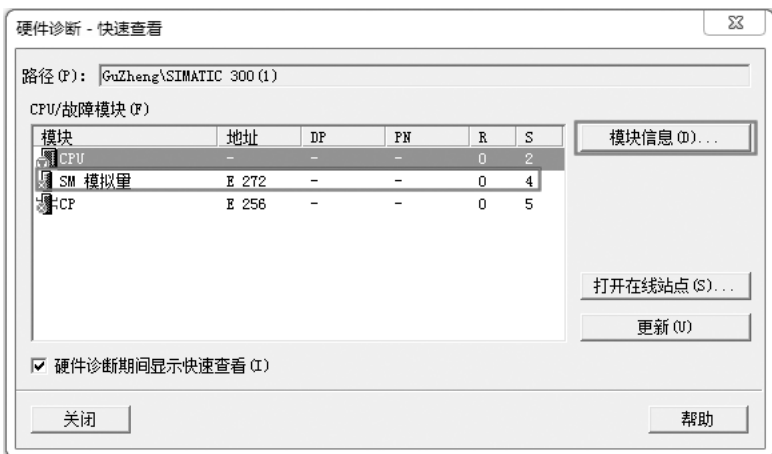



图 12-13　硬件诊断 - 快速查看

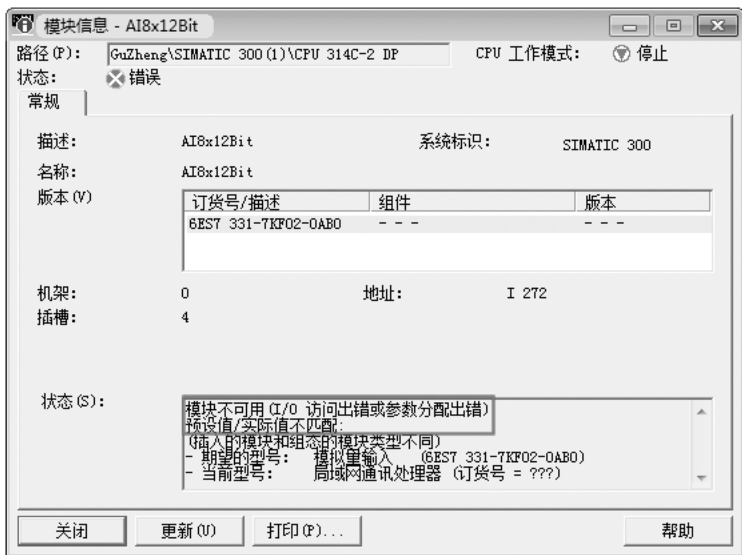



图 12-14　模块信息 - 状态区

（5）模块的断线故障

在 SIMATIC Manager 界面中，单击“PLC”→“诊断/设置”→“硬件诊断”。可以看到 CPU 模块处于“STOP”状态，SM 和 CP 模块出错，如图 12-15 所示。选中“SM 模拟量”模块，单击“模块信息”按钮，弹出如图 12-16 所示的界面。

如图 12-16 所示，在“诊断中断”选项卡中，可以看到模拟量模块的通道 0 和通道 1 断线。从图 12-15 可以看到此模拟量模块的起始地址是 PIW272，那么通道 0 和通道 1 对应的模拟量的地址为 PIW272 和 PIW274，维修时检查这两个通道即可。

（6）通信和机架故障

在 SIMATIC Manager 界面中，单击“PLC”→“诊断/设置”→“硬件诊断”。可以看到 DP 从站上有个红色斜线，这是掉站的标志，属于通信故障，如图 12-17 所示。

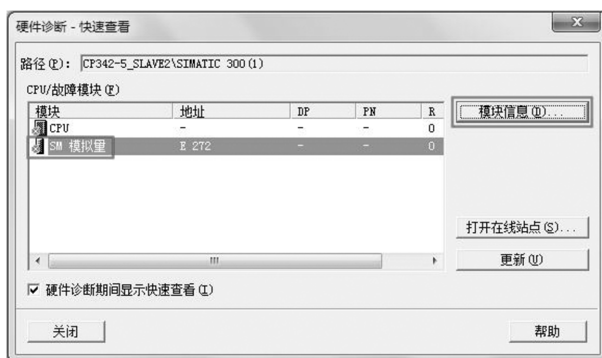



图 12-15　硬件诊断 - 快速查看




图 12-16　模块信息 - 状态区

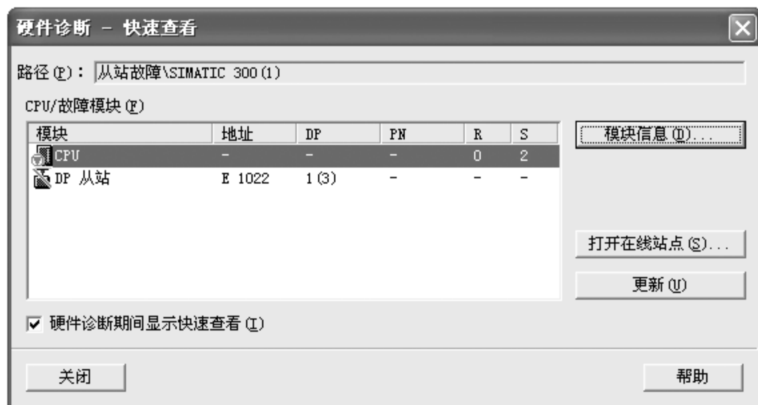



图 12-17　硬件诊断 - 快速查看

单击“DP 从站诊断”选项卡，可以看到，故障信息：“DP 从站不能访问总线”，如图 12-18 所示。虽然确定故障是从站通信故障，但仍然需要进一步查找通信故障产生的具体原因。

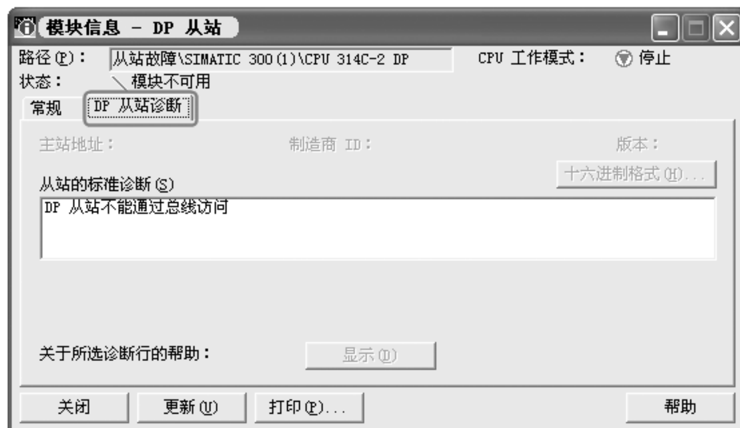



图 12-18　模块信息 - DP 从站

查看“诊断缓冲区”，可以看到分布式 I/O 故障，具体是分布式机架故障造成的 I/O 故障，即从站出现故障，如图 12-19 所示，它是机架故障造成的。机架损坏和机架断电都报

机架故障，检查后发现故障原因为机架断电。




图 12-19 模块信息 - 诊断缓冲区

在实际工作中，需要通过表层的故障现象一步步查找出深层次的故障原因，本例的机架断电造成机架故障，再造成通信故障。在“诊断信息”中看到的通信错误和机架错误都是表层的故障现象，而分布式机架的断电才是故障的根源之所在，但表层的“诊断信息”提供了很好的诊断依据，是特别有用的。

也可以在硬件组态界面中查看故障站点。方法如下：

在 SIMATIC 管理器中，单击“在线”按钮，再双击“硬件”，打开硬件组态界面，即可诊断，如图 12-20 所示。也可以在硬件组态界面中直接单击“在线”按钮。很明显 3 号站上有红色斜线，表明 3 号站故障（掉站）。

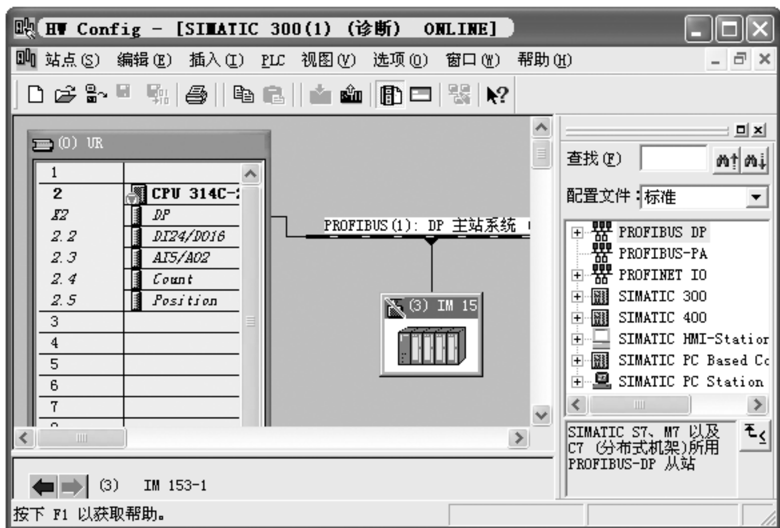



图 12-20 硬件组态界面中查看故障站点

（7）S7 -400 的外部错误

在 SIMATIC Manager 界面中，单击“PLC”→“诊断/设置”→“硬件诊断”，如图 12-21 所示。选中“CPU”模块，单击“模块信息”按钮，弹出如图 12-22 所示界面。

在图 12-22 的“常规”选项卡中，可以看到“EXTF”错误，也就是外部错误。再打开“诊断缓冲区”选项卡，如图 12-23 所示，可以看到 CPU 的外部错误是电池故障，电池故障一般是 S7 -400 没安装电池或者电池电压过低。




图 12-21　硬件诊断 - 快速查看

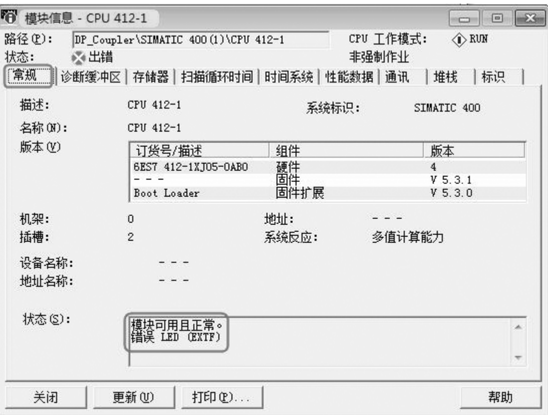



图 12-22　模块信息 - 常规

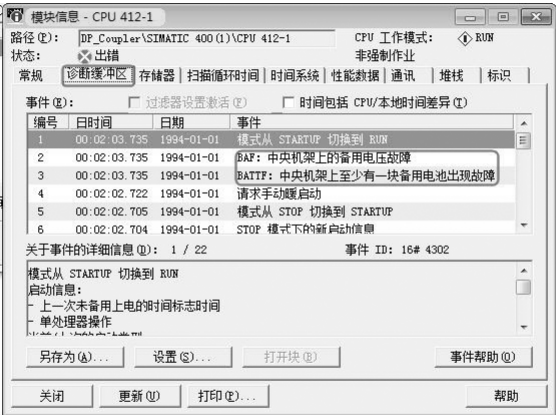



图 12-23　模块信息 - 诊断缓冲区

## 12.2.3　用通信块（如 SFB15）的输出参数/返回值（RET_ VAL）诊断故障

利用输出参数诊断故障，梯形图如图 12-24 所示，当 M30.0 =1，表示有错误。例如，当 MW40 =1 表示通信错误；MW40 =25 表示通信已起动，正在处理等。其余故障信息可以参见表 12-8。

**表 12-8　故障信息表**

| 序号 | ERROR | STATUS（十进制） | 含　　义 |
|---|---|---|---|
| 1 | 0 | 11 | 警告：<br>● 由于前一个作业还没有结束，所以不能执行新作业<br>● 正在以较低优先级处理此作业 |
| 2 | 0 | 25 | 通信已经起动。作业正在处理 |
| 3 | 1 | 1 | 通信故障<br>● 没有装载连接描述（本地或远程）<br>● 连接被中断（例如：电缆断线、CPU 关闭、或 CP 处于 STOP 模式）<br>● 没有建立到通信伙伴的连接<br>● FB 无法在 S7 -400 CPU 上运行<br>● 此外对于 S7 -300，超出并行作业/背景的最大数目 |
| 4 | 1 | 2 | 来自伙伴设备的否定确认。无法执行该函数 |

【例 12-1】某系统由 CPU314C－2DP、CP342－5 和 IM153－1 组成，其中 CP342－5 是 DP 主站，调试时，监控发现如图 12-25 所示程序，分析原因。

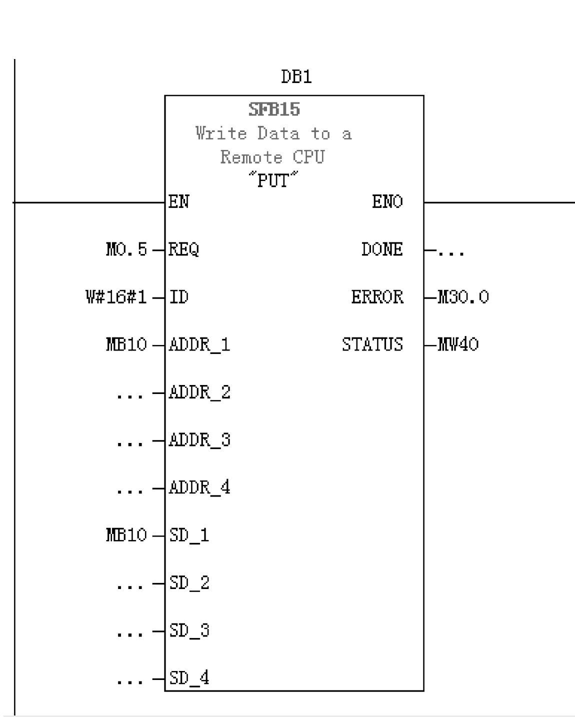



图 12-24　梯形图

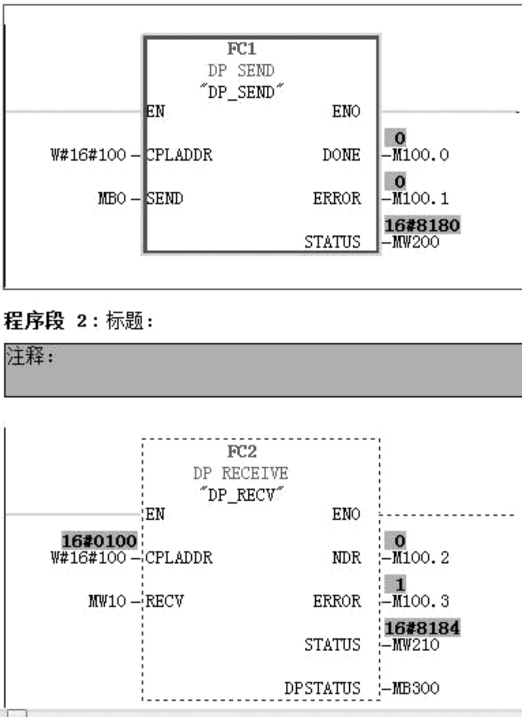



图 12-25　梯形图

**解：**

1）监控程序，发现发送块 FC1 的 ERROR＝0，可见发送是正常的，而接收块 FC2 的 ERROR＝1，接收不正常。

2）再查看帮助，发送块 FC1 的 STATUS＝16#8180，其含义是激活数据传送，进一步说明发送正常。接收块 FC2 的 STATUS＝16#8184，其含义是系统错误或非法参数类型，进一步说明接收不正常。

3）能发送，不能接收，基本排除硬件错误。但故障并没有诊断出来，此时结合前面章节的“诊断视图”分析，打开“诊断缓冲区”如图 12-26 所示。可见：写入发生区域长度错误，在“MB300”处。

4）将程序中的 MB300 改为 MB200，错误排除，接收正常。




图 12-26　模块信息－诊断缓冲区

可见：进行故障诊断时，有时将几种诊断方法联合使用效果会更好。

### 12.2.4　用组织块进行故障诊断

S7－300/400 PLC 的系统错误分为同步错误和异步错误。

同步错误是与程序执行有关的错误，对应的组织块是 OB121 和 OB122。

异步错误是与 PLC 的硬件或操作系统密切相关的错误，与程序执行无关，异步错误对应的组织块见表 12-9（除去 OB121 和 OB122 的组织块外，其余的组织块）。

当 CPU 检测到错误时，会调用适当的组织块，见表 12-9。如果没有相应的错误处理 OB，CPU 将进入 STOP 模式。用户可以在错误处理 OB 中编写如何处理这种错误的程序，以减小或消除错误的影响。

**表 12-9　错误处理组织块**

| OB 号 | 错 误 类 型 | 优　先　级 |
|---|---|---|
| OB70 | I/O 冗余错误（仅 H 系列 CPU） | 25 |
| OB72 | CPU 冗余错误（仅 H 系列 CPU） | 28 |
| OB73 | 通信冗余错误（仅 H 系列 CPU） | 35 |
| OB80 | 时间错误 | 26 |
| OB81 | 电源错误 | 26/28 |
| OB82 | 诊断中断 | |
| OB83 | 插入/取出模块中断 | |
| OB84 | CPU 硬件故障 | |
| OB85 | 优先级错误 | |
| OB86 | 机架故障或分布式 I/O 的站故障 | |
| OB87 | 通信错误 | |
| OB121 | 编程错误 | 引起错误的 OB 的优先级 |
| OB122 | I/O 访问错误 | |

用组织块诊断故障的原理就是当 CPU 发生故障时，调用对应的组织块，组织块中编写有对应的程序，提供故障诊断的信息，例如当模拟量模块 SM331 的某一通道的传感器断线了，调用 OB82 中的程序（假设编程序已经编写了），从而通知 HMI，哪一个通道断线，这样维修工程师只要查看触摸屏就很容易查找故障了。以下用几个例子介绍 OB 块诊断故障的方法。

**【例 12-2】**控制系统为 S7－300 和 SM331，当传感器采集的电流大于 16 mA 时报警，小于此值时停止报警，用 OB82 实现此功能。

**解：**

1）先进行硬件组态。要使用 OB82 诊断故障，SM331 模块必须有诊断功能。硬件组态过程如下：

系统硬件组态如图 12-27 所示，双击“AI8x12Bit”，弹出如图 12-28 所示的界面，配置 SM331 模块的“输入（Inputs）”选项，选择 0－1 通道组为 4 线制电流（4DMU），其他通道组为“取消激活”，并注意模块的量程卡要与设置的相同。选中“启用（Enable）”框中的“诊断中断（Diagnostic Interrupt）”选项，选中“诊断（Diagnostics）”选项中的 0－1 通道组

中的“组诊断（Group Diagnostics）”选项，并在“上限框”中输入 16 mA，最后单击“确定”按钮。




图 12-27　硬件组态（1）



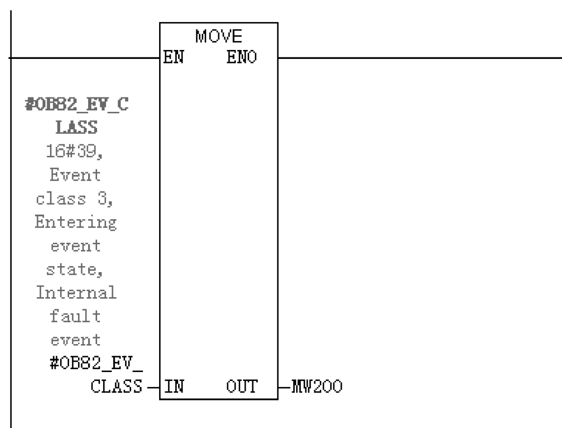

图 12-28　硬件组态（2）

2）编写梯形图程序。在编写梯形图程序之前，先要掌握 OB82 两个变量的含义。

OB82_EV_CLASS 是事件级别和标识（见表 5-9），当其值为 B#16#38（56），表示事件离去；当其值为 B#16#39（57），表示事件到来，对应本例，就是当传感器采集到的电流大于 16 mA 时，表示事件到来，而当传感器采集到的电流小于 16 mA 时，表示事件离开，理解这点很重要。

OB82_MDL_ADDR 就是故障发生处模板的逻辑起始地址，即模拟量模块的通道起始地址。

编写梯形图程序如图 12-29 所示。



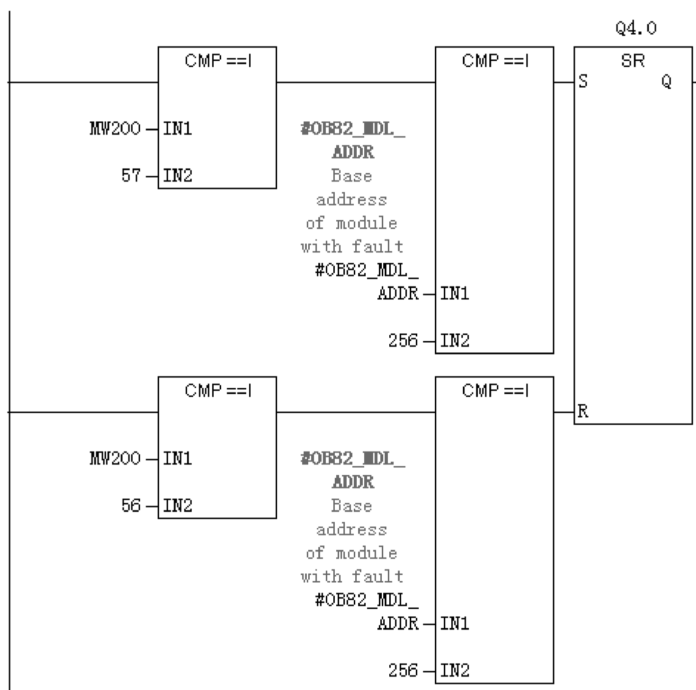




图 12-29　梯形图

3）运行仿真。从理论上讲，在仿真器的 PIW256 中输入大于 22119 的数字量（此数字量对应 16mA 的模拟量），则 Q4.0 置位，而在仿真器的 PIW256 中输入小于 22119 的数字量（此数字量对应 16 mA 的模拟量），则 Q4.0 复位，但实际这么做不会有以上效果。正确的做

法是，先把完整的程序下载到仿真器中，将仿真器置于“RUN”状态，在工具栏中单击“Execute”（执行）→“Trigger Error OB”（触发器）→“Diagostic Interrupt”（诊断中断），如图 12-30 所示，打开诊断中断界面。

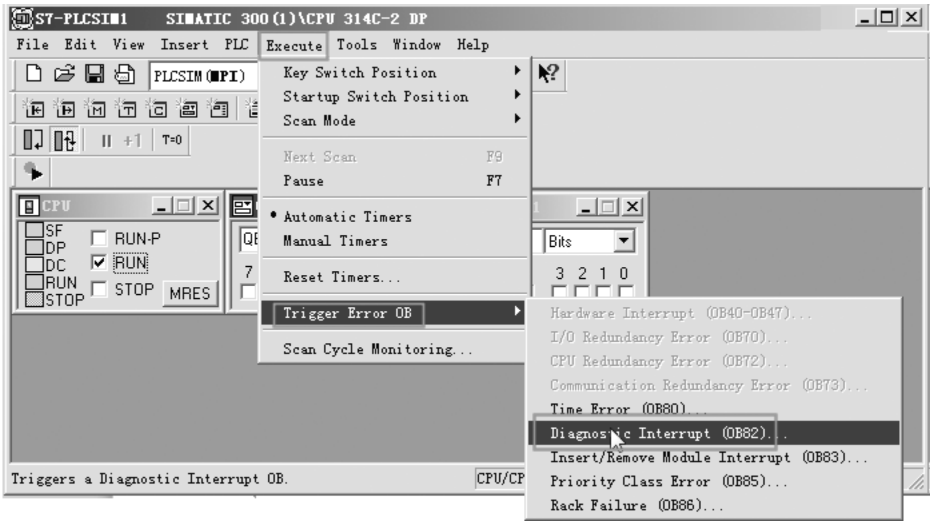


图 12-30　打开“诊断中断”界面

如图 12-31 所示，在“Module address”（模块地址）中输入故障模块的逻辑地址（即模拟量的通道号）“piw256”，任意选择一个可能的故障，本例选择“Channel fault”（通道故障），单击“Apply”（应用）按钮，仿真器中的 Q4.0 亮，如图 12-32 所示。同理，当去掉所有故障选项，再单击“Apply”（应用）按钮时，仿真器中的 Q4.0 灭。

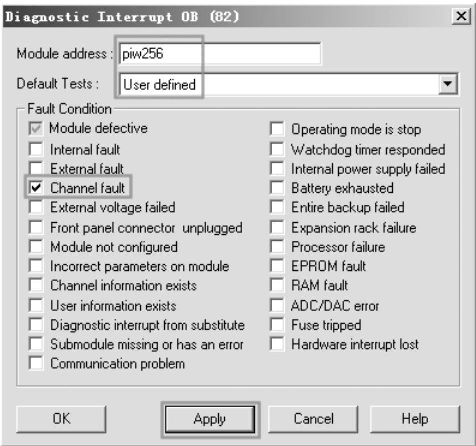


图 12-31　设置故障



图 12-32　故障显示

【关键点】要利用 OB82 编写故障诊断的程序相对较难，关键要掌握组织块 OB82 的各项参数的含义。此题用指令表编写更加简洁，如下所示：

此题用指令表编写更加简洁，如下所示：

```
        SET
        SAVE
        =       L     20.1
        L       #OB82_EV_CLASS
        L       B#16#39
        ==I
        JCN     MOO1
        SET
        =       Q     4.0
        JU      MOO2
  MOO1: L       #OB82_EV_CLASS
        L       B#16#38
        ==I
        JCN     MOO2
        CLR
        =       Q     4.0
  MOO2: CL4
        A       L     20.1
        SAVE
        BE
```

**【例 12-3】**控制系统为 S7－300、ET200、SM323 和 SM332，SM332 输出电压信号。当电压信号断线时，能通过 STEP 7 的参数表查看到故障信号的通道，要求用 OB82 实现此功能。

**解：**

1）硬件组态，过程如下：

首先按照如图 12-33 进行组态，双击“AO2＊12bIT”，弹出如图 12-34 所示的界面，在“输出”选项卡中，勾选“组诊断”和“诊断中断”，最后单击“确定”按钮。

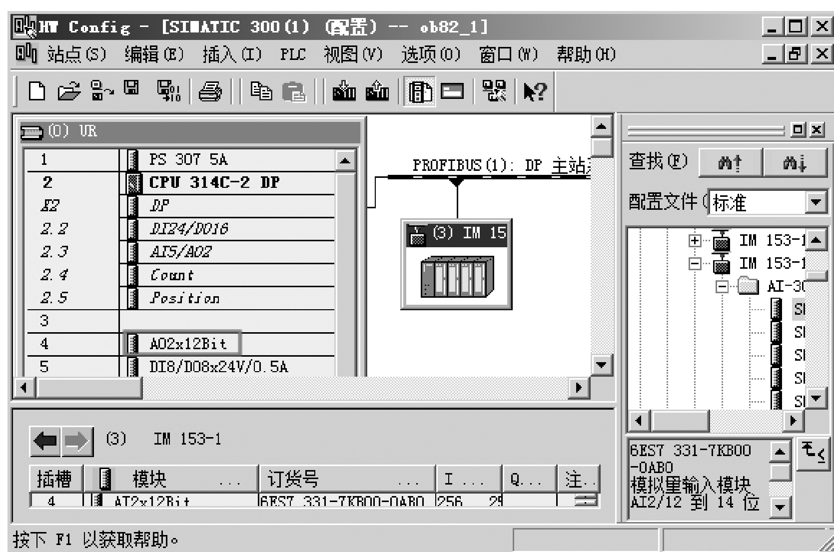



图 12-33 硬件组态（1）

2）创建数据块 DB1，并在 DB1 中创建一个数组 ary[0..20]，数组的容量为 21 B，如图 12-35 所示。

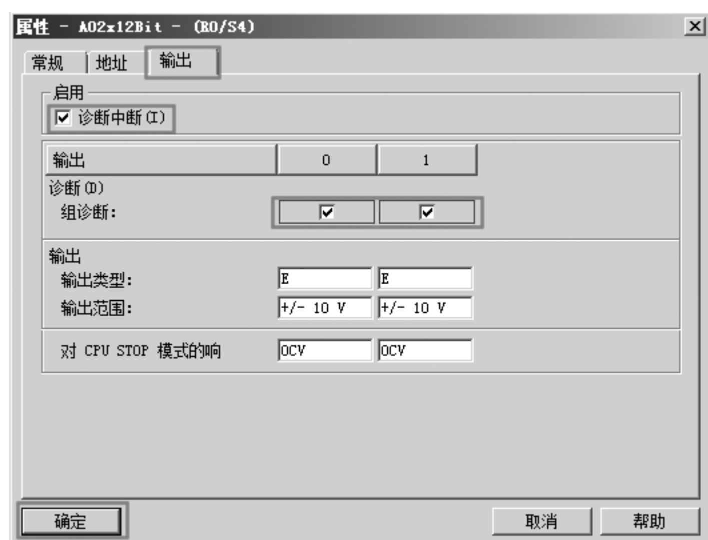


图 12-34　硬件组态（2）



图 12-35　创建的数组

3）创建变量表 VAT_1，并把需要监控的参数输入变量表，如图 12-36 所示，利用这个变量可以监控到故障信息。



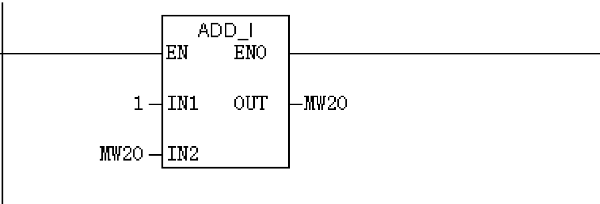
图 12-36　创建变量表

4）编写梯形图程序。在 OB82 中编写梯形图程序，如图 12-37 所示。由于 S7－300/400

有很强的自诊断功能，故障信息自动赋值给 OB82 的参数。程序的作用实际就是将 OB82 的所有参数信息传送到数组 DB1. ary 中，这样故障信息就可以在参数表中显示了。

**程序段 1：标题：**

故障次数记录




**程序段 2：标题：**

将OB82的所有参数信息传送到数组DB1. ary中，之后可以在参数表中显示

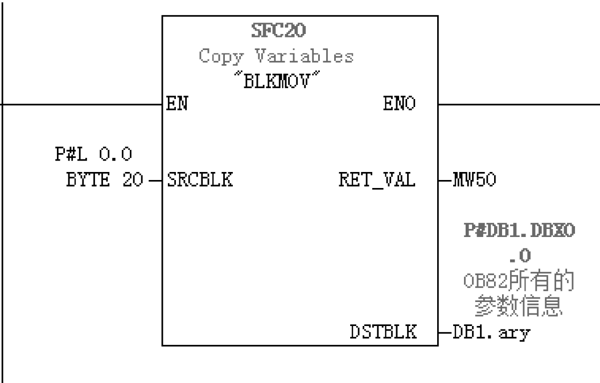



图 12-37　梯形图

5）仿真。先把完整的程序下载到仿真器中，将仿真器置于“RUN”状态，在工具栏中单击“Execute”（执行）→“Trigger Error OB”（触发器）→“Diagostic Interrupt”（诊断中断），如图 12-38 所示，打开诊断中断界面。

如图 12-39 所示，在“Module address”（模块地址）中输出故障模块的逻辑地址（即模拟量的通道号）“pqw256”，任意选择一个可能的故障，本例选择“Exernall fault”（外部故障），单击“OK”（确定）按钮，模拟外部断线。

6）监视故障。打开变量表，单击“监视变量”按钮，可以看到参数，如图 12-40 所示。参数的含义如下：

① MW20 代表故障的累积次数。

② DB1. DBB0 为 16#39，表示事件到来。

③ DB1. DBB5 为 16#55，表示为输出模块。

④ DB1. DBW6 为 16#0100，就是输出模块的地址是 PQW256（256 的十六进制就是 16#0100）。

⑤ DB1. DBW8 为 16#1105（2#0001 0001 0000 0101），表示外部故障、通道故障和模块

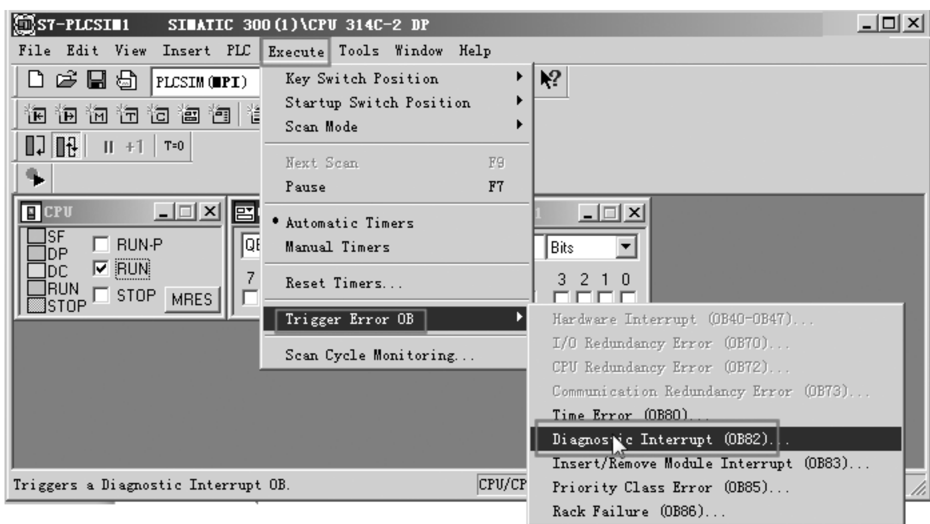



图 12-38　打开“诊断中断”界面

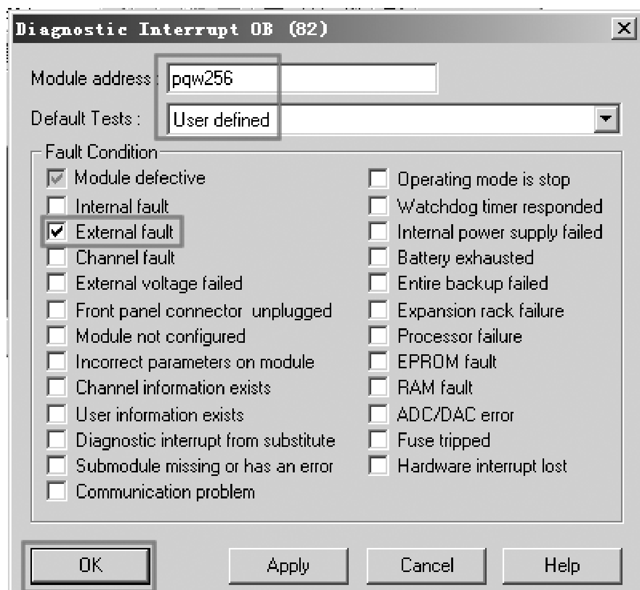



图 12-39　模拟故障

发生故障。

**【例 12-4】** 控制系统为 S7－300、ET200、SM323 和 SM332，SM332 输出电压信号。当电压信号断线时，能通过 STEP 7 的参数表查看到故障信号的通道，要求用 OB86 实现此功能。

硬件组态与上例相同，OB86 中的梯形图程序如图 12-41 所示。

创建如图 12-42 所示的变量表，并监控。

图 12-42 中变量表的含义如下：

1）DB1. DBB0 为 16#39，表示事件到来。

变量 - VAT_1

表格(T) 编辑(E) 插入(I) PLC 变量(A) 视图(V) 选项(O) 窗口(W) 帮助(H)

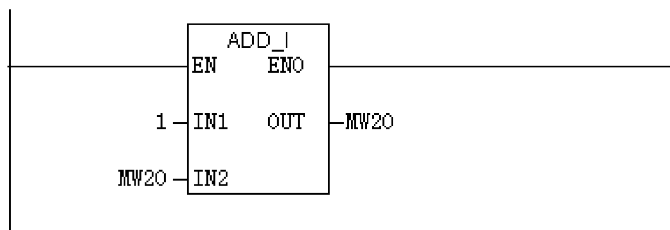
VAT_1 -- @ob82_1\SIMATIC 300(1)\CPU 314C-2 DP\S7 程序(1) ON...

| | 地址 | 符号 | 显示格式 | 状态值 | 修改数值 |
|---|---|---|---|---|---|
| 1 | MW 20 | | HEX | W#16#0001 | |
| 2 | DB1.DBD 0 | | HEX | DW#16#39421A52 | |
| 3 | DB1.DBD 4 | | HEX | DW#16#C5550100 | |
| 4 | DB1.DBD 8 | | HEX | DW#16#11050000 | |
| 5 | DB1.DBD 12 | | HEX | DW#16#15020713 | |
| 6 | DB1.DBD 16 | | HEX | DW#16#22427847 | |
| 7 | | | | | |

ob82_1\SIMATIC 300(1)\...\S7 程序(1) RUN Abs

图 12-40 监视故障

**程序段 1：标题：**




**程序段 2：标题：**

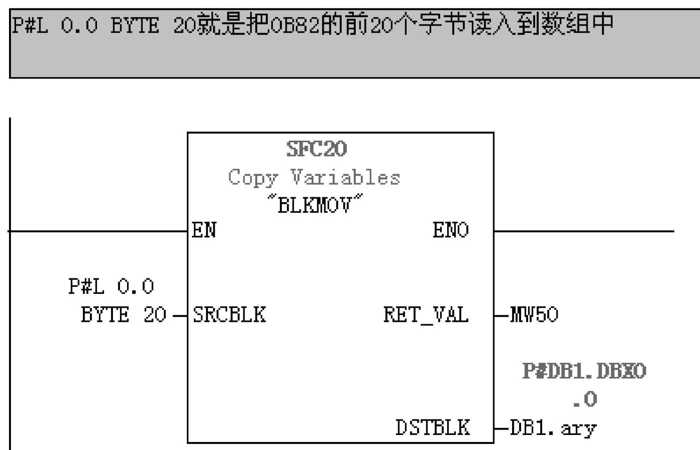



图 12-41 梯形图

2）DB1. DBB1 为 16#C4，表示为从站故障。

3）DB1. DBB2 为 16#1A，表示 OB 优先级是 26。

4）DB1. DBB3 为 16#56，表示 OB 编号是 86。

5）DB1. DBW8 为 16#03FE，是从站诊断地址。

6）DB1. DBW10 为 16#0103，表示主站系统编号是 01，从站是 03。

| | 地址 | 符号 | 显示格式 | 状态值 | 修改数值 |
|---|---|---|---|---|---|
| 1 | MW 20 | | DEC | 1 | |
| 2 | MW 50 | | DEC | 0 | |
| 3 | DB1.DBD 0 | | HEX | DW#16#39C41A56 | |
| 4 | DB1.DBD 4 | | HEX | DW#16#C05403FF | |
| 5 | DB1.DBD 8 | | HEX | DW#16#03FE0103 | |
| 6 | DB1.DBD 12 | | HEX | DW#16#13091515 | |
| 7 | DB1.DBD 16 | | HEX | DW#16#17544141 | |
| 8 | | | | | |

图 12-42　变量表

7）DB1. DBD12 和 DB1. DBD16 表示事件发生的时间是 13 年 9 月 15 日 15 时 17 分 54 秒。

注意：DB1. DBD0 包含 DB1. DBB0、DB1. DBB1、DB1. DBB2 和 DB1. DBB3 共四个字节。

诊断地址和 DP 站地址是不同的概念。选中 DP 网线，如图 12-43 所示，1022 就是 16# 03FE。选中从站 IM153 - 1 上的 PROFIBUS，即可看到从站诊断地址。

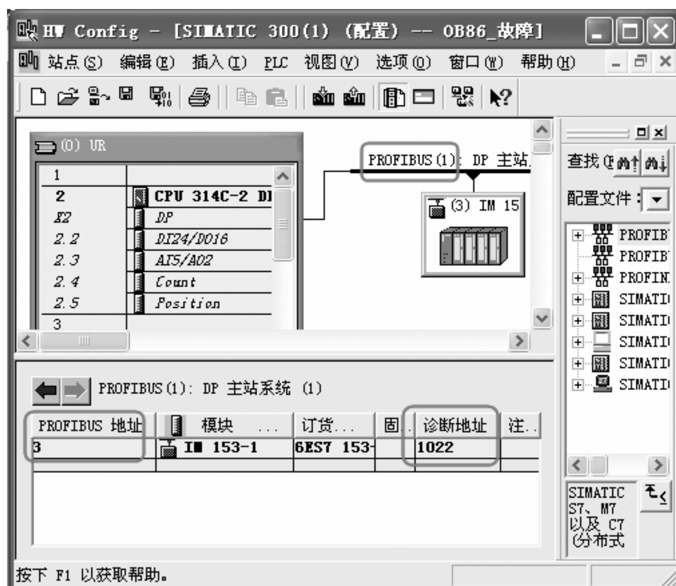



图 12-43　诊断地址

## 12. 2. 5　用 SFC13/SFC51/FC10/FB10/FC3 进行故障诊断

### 1. 用 SFC13 进行故障诊断

SFC13 是 STEP 7 中专用的诊断指令，具有强大的诊断功能。

以下用一个例子介绍 SFC13 的应用。先进行如图 12-44 所示硬件组态。

双击图 12-44 所示的从站模块 IM153 - 1，弹出的界面可以看到从站的诊断地址 1022（就是 16#03FE），如图 12-45 所示。

在 DB82 中创建数组，如图 12-46 所示。

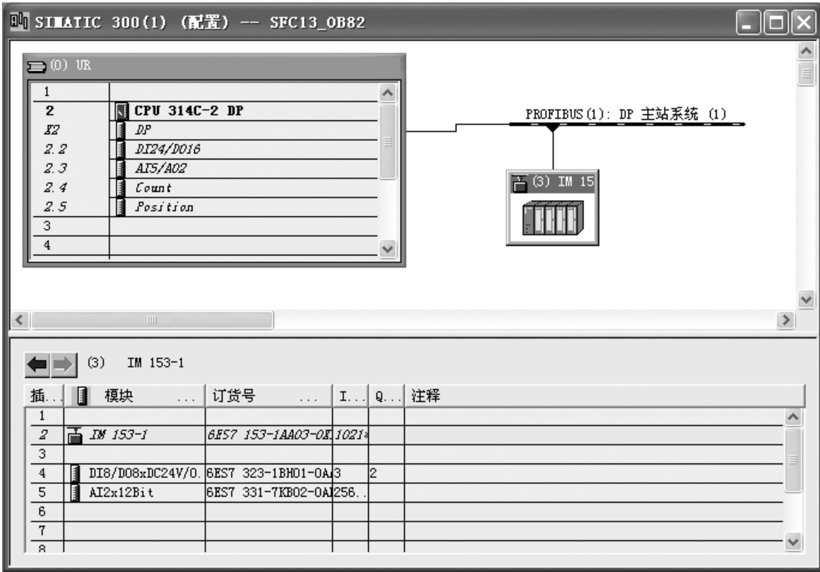



图 12-44　硬件组态




图 12-45　DP 从站属性

在 OB82 中编写程序，如图 12-47 所示。

创建并监控变量表如图 12-48 所示，以下解读故障含义。

如未编写如图 12-47 所示的程序，单击如图 12-49 中的“十六进制格式”按钮，弹出从站信息，此信息与变量表中的信息的含义是相同的。

在解读十六进制代码的含义前，须知：这些代码和不同的从站设备是相关的，读者应查阅相关手册，得出代码表达的含义。本例使用的是 ET－200M，所以读者要查阅 ET－200M 手册。以下给出部分手册查阅到的信息。ET200M 的诊断数据结构见表 12-10。

```
程序段 1：标题：

      L     MW    80
      +     1
      T     MW    80

程序段 2：标题：

      L     B#16#39
      L     #OB82_EV_CLASS
      ==I
      JC    m002

m001: CALL  "DPNRM_DG"
       REQ    :=TRUE
       LADDR  :=W#16#3FE
       RET_VAL:=MW52
       RECORD :=P#DB82.DBX32.0 BYTE 29
       BUSY   :=M50.1
      A     M     50.1
      JC    m001
      BEU

m002: CALL  "DPNRM_DG"                    SFC13
       REQ    :=TRUE
       LADDR  :=W#16#3FE
       RET_VAL:=MW54
       RECORD :=P#DB82.DBX0.0 BYTE 29
       BUSY   :=M50.2
      A     M     50.2
      JC    m002
```

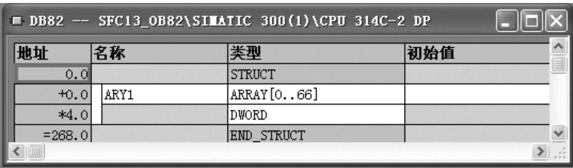



图 12-46　创建数组

图 12-47　程序

VAT_1 -- @SFC13_OB82\SIMATIC 300(1)\CPU 314C-2 DP\S7 程序(1)...

| | 地址 | | 符号 | 显示格式 | 状态值 | 修改数值 |
|---|---|---|---|---|---|---|
| 1 | DB82.DBD | 0 | | HEX | DW#16#00040002 | |
| 2 | DB82.DBD | 4 | | HEX | DW#16#418B0000 | |
| 3 | DB82.DBD | 8 | | HEX | DW#16#00000000 | |
| 4 | DB82.DBD | 16 | | HEX | DW#16#00000000 | |
| 5 | DB82.DBD | 20 | | HEX | DW#16#00000000 | |
| 6 | DB82.DBD | 24 | | HEX | DW#16#00000000 | |
| 7 | DB82.DBD | 28 | | HEX | DW#16#00000000 | |
| 8 | | | | | | |
| 9 | DB82.DBD | 32 | | HEX | DW#16#00000000 | |

SFC13_OB82\SIMATIC 300(1)\...\S7 程序(1)　RUN　Abs < 5.2

图 12-48　变量表

**表 12-10　ET200M 的诊断数据结构**

| 字 节 号 | 意　义 | 字 节 号 | 意　义 |
|---|---|---|---|
| 0 ~ 2 | 站状态 | 6 ~ 8 | 与标识符有关的诊断数据 |
| 3 | PROFIBUS 地址 | 9 ~ 28 | 中断（站诊断），最多 20B |
| 4 ~ 5 | 制造商 ID | X ~ X + 7 | H 状态，仅用于 S7 - 400 |

站状态 1 各位为 1 的意义见表 12-11。




图 12-49　故障代码

**表 12-11　站状态 1 各位为 1 的意义**

| 位 | 意　　义 |
|---|---|
| 0 | DP 主站不能访问 DP 从站 |
| 1 | DP 从站尚未准备好进行数据交换 |
| 2 | DP 主站发送到 DP 从站的组态数据与 DP 从站的实际组态不匹配 |
| 3 | 有外部诊断信息 |
| 4 | DP 从站不支持请求功能 |
| 5 | DP 主站不能解释 DP 从站响应 |
| 6 | DP 从站类型与软件组态不相符合 |
| 7 | 其他 DP 主站（不是当前访问 DP 从站的 DP 主站）已组态 DP 从站 |

站状态 2 各位为 1 的意义见表 12-12。

**表 12-12　站状态 2 各位为 1 的意义**

| 位 | 意　　义 |
|---|---|
| 0 | 必须重启 DP 从站 |
| 1 | 有诊断消息，在解决问题之前，DP 从站不会运行 |
| 2 | DP 从站的该位始终为 1 |
| 3 | 已启用 DP 从站的响应监视器 |
| 4 | DP 从站已接收到冻结（FREEZE）控制命令 |
| 5 | DP 从站已接收到同步（SYNC）控制命令 |
| 6 | 该位始终为 0 |
| 7 | DP 从站被禁用，即已将从当前处理中隔离出来 |

站状态 3 一般为 16#00。

如图 12-50 所示，第 0 个字节 16#00、第 1 个字节 16#04 和第 2 个字节 16#00 表示无故障，第 3 个字节 16#02 表示主站地址是 2，第 4、5 个字节 16#418B 是制造商的 ID，这个代号可以在“模块信息”中看到。




图 12-50　故障代码

**2. 用 SFC51 进行故障诊断**

系统功能 SFC51 “RDSYSST”（读取系统状态）可以读取系统状态列表或部分系统状态列表。用读取的信息进行故障诊断。

（1）系统状态列表（SSL）介绍

1）系统状态列表（SSL）的作用。系统状态列表（SSL）用于描述 PLC 的当前状态。SSL 的内容只能通过系统功能进行读取，而不能修改。换言之，部分列表是虚拟列表，只在有特殊请求时由 CPU 的操作系统所创建。

系统功能 SFC 51 “RDSYSST” 用于读取 “系统状态列表”、部分列表或 CPU 的 SSL 列表摘录。对于 PROFINET IO，这些 SSL 包含了 I/O 模块、PROFINET IO 主站系统或实际控制器的状态信息。当选择所用的、特别是在一个中断或起动 OB 中使用的 SSL_ID 时，必须注意 SFC 仅能同步执行。如果执行 SFC 之后，Busy 位激活，表明几个循环周期执行一次 SFC，数据还没有完全读出，因此数据是无效的。接收到的系统状态列表数据记录包括了诊断 PROFINET IO 设备上的信息概览。

PROFINET 通信部分列表是虚拟列表，只是在有请求时由 CPU 的操作系统所创建。虚拟列表来自 CPU 内部的 PROFINET IO 控制器的缓冲区，缓冲区由控制器的起动和 ALARM 信息进行维护和刷新。

2）局部系统状态列表（SSL）的结构。局部系统状态列表（SSL）由报头和数据记录组成。报头由局部 SSL 的标识符 SSL_ID、索引 INDEX、包含局部 SSL 中的数据记录的字长度和数据记录的个数组成。局部 SSL 的标识符 SSL_ID 占一个字（16 位），它由局部 SSL 编号（第 0 ~ 7 位）、局部 SSL 摘录号（第 8 ~ 11 位）和模块类型（第 12 ~ 15 位）组成，如图 12-51 所示。

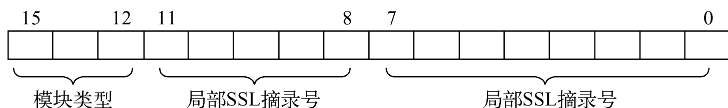



图 12-51　局部 SSL 的标识符 SSL_ID

SSL_ID 的模块类型占高四位，主要读取模块的类型，如果读取到的二进制数据为 2#0000、2#0100、2#1000 和 2#1100，则分别对应的模块类型为 CPU 模块、接口模块 IM、功能模块 FM 和通信模块 CP。

某些局部 SSL 或者它的摘录需要一个对象类型标识符或者一个对象号，此时必须使用索引 INDEX。通过局部 SSL 的摘录号，可以指定要读取局部列表的哪一个子集。部分 SSL_ID 列表和 INDEX 的对应关系见表 12-13。

**表 12-13　部分 SSL_ID 列表和 INDEX 的对应关系**

| SSL_ID（W#16#...） | 部分列表（PROFINET） | INDEX（W#16#...） |
| --- | --- | --- |
| 0C91 | PROFINET 接口模块（集成的或外部的）上模块的状态信息 | 逻辑基地址，输出模块 bit15 = 1 |
| 0D91 | PROFINET 中所有模块的状态信息 | bits 0 ~ 10：站号，bits 11 ~ 14：PROFINET IO 系统总线号的后两位，Bit 15:1 |
| 0094 | IO 控制器系统的中央机架/站中的机架的期望状态 | 100 ~ 115：PROFINET IO 系统总线号 |
| 0294 | IO 控制器系统的中央机架/站中的机架的实际状态 | 同上 |
| 0694 | IO 控制器系统的中央机架/站中的扩展单元的诊断状态 | 同上 |
| 0794 | IO 控制系统的中央机架/站中的机架的维护状态 | 同上 |
| 0C96 | PROFINET 中所有模块的状态信息 | 逻辑基地址，输出模块 bit15 = 1 |
| 0091 | 所有插入模块/子模块的状态信息 | 与 INDEX 无关 |
| 0391 | 所有无法获得的模块的状态信息 | 与 INDEX 无关 |

（2）SFC51 指令介绍

读取“系统状态列表（SFC51）指令比较难理解，其含义见表 12-14。

**表 12-14　读取“系统状态列表（SFC51）指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存　储　区 |
| --- | --- | --- | --- | --- |
| “RDSYSST”<br>EN　ENO<br>REQ　RET_VAL<br>SZL_ID　BUSY<br>INDEX　SZL_HEADER<br>DR | EN | BOOL | 使能输入端 | |
| | REQ | BOOL | REQ = 1 时起动处理 | I、Q、M、D、L、常数 |
| | SZL_ID | WORD | 读取的系统状态列表或部分列表的 SSL – ID | I、Q、M、D、L 、常数 |
| | INDEX | WORD | 部分列表中对象的类型或编号 | I、Q、M、D、L、常数 |
| | BUSY | BOOL | 为 TRUE 时尚未完成读取 | I、Q、M、D、L |
| | SZL_HEADER | STRUCT | 结构：详细参见下文 | |
| | DR | ANY | SSL 列表读取或 SSL 部分列表读取的目标区域 | I、Q、M、D、L |

要理解这条指令，必须对照表 12-13，或者查看帮助。例如，当 SZL_ID = 16#0694 和 INDEX 为 PROFINET IO 系统总线号（取值范围是 100 ~ 115，如图 12-53 中为 100）配合时，读取的信息是 IO 控制器系统的中央机架/站中的扩展单元的诊断状态。有时，SZL_ID 的数

值与 INDEX 数值无关，例如当 SZL_ID = 16#0391 时，读取的信息是所有无法获得的模块的状态信息，不需要指定 INDEX 数值，即与 INDEX 数值无关。

SFC51 的 SZL_HEADER 的数据类型是 STRUCT（结构），必须在调用 SFC51 的组织块中创建结构。创建的方法是在 OB 块的局部变量的最下方生成临时变量 SZL_HEADER，其数据类型为 STRUCT。本例是在 OB1 中创建，局部变量表的结构如图 12-52 所示。SZL_HEADER 中有两个元素，分别为 LENTHDR 和 N_DR，其数据类型为 WORD（字）。LENTHDR 是读取的 SSL 或者 SSL 的数据记录的长度，N_DR 是读取的数据记录的编号。

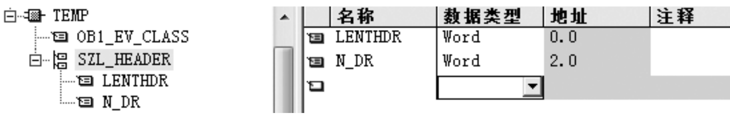



图 12-52　局部变量表的结构

（3）SFC51 的应用举例

**【例 12-5】**某控制系统采用 PROFINET 通信，其硬件组态如图 12-53 所示，请用 SFC51 指令读取 PROFINET IO 的故障信息。

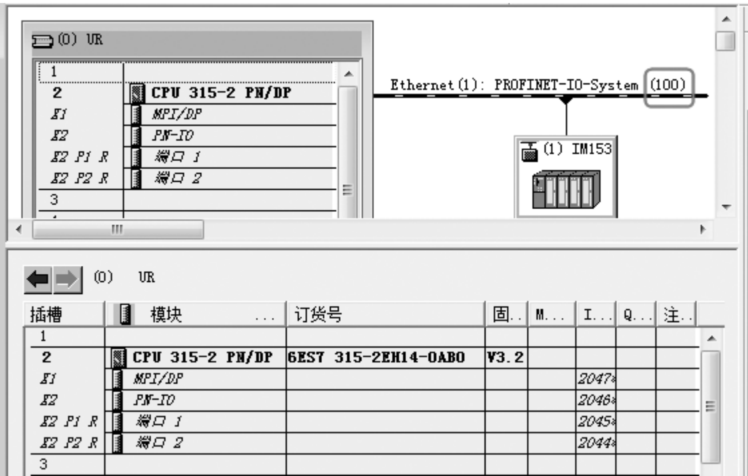



图 12-53　系统硬件组态

**解：**

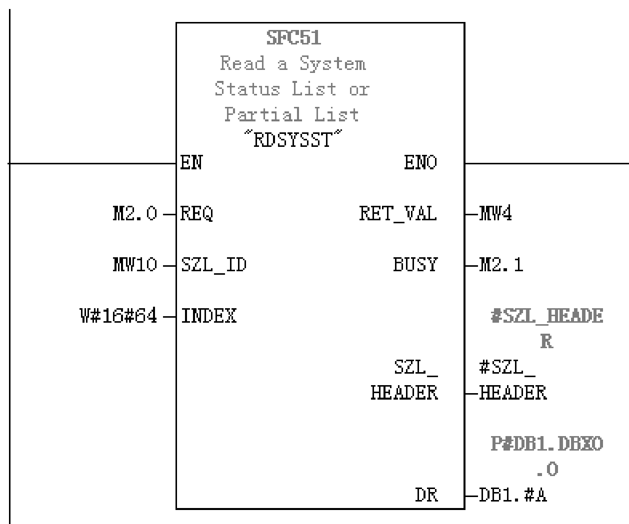
1）先创建共享数据块 DB1，用于存储故障信息。

2）再在 OB1 中编写如图 12-54 所示的梯形图程序，注意编写程序之前，需要创建如图 12-52 所示的结构 SZL_HEADER，否则编写的程序会报错。

注意本例的 INDEX = 16#64 就是十进制的 100，也就是如图 12-53 中 PROFINET IO 系统总线号，理解这点十分重要。

3）创建变量表进行监控，如图 12-55 所示。MW6 中是数据记录的长度，监控为 258，MW6 为数据记录个数，监控为 1。MW10 = 16#0294 是 SSL_ID。

对于 DB1. DBB6 为 2#0000_0011，其中 bit0 表示组信息，如果为 1，表示至少有一个 IO 设备与 IO 控制器进行通信；如果为 0，表示 IO 站点全部丢失。Bit1 ~ Bit5 表示设备号 1 ~ 5




⊟ **程序段 2**：标题：




图 12-54　梯形图

变量 - [VAT_1 -- @SFC51\SIMATIC 300(1)\CPU 315-2 PN/DP\S7 程序(1)  ONLINE]

表格(T)　编辑(E)　插入(I)　PLC　变量(A)　视图(V)　选项(O)　窗口(W)　帮助(H)

| | 地址 | 符号 | 显示格式 | 状态值 | 修改数值 |
|---|---|---|---|---|---|
| 1 | MW 6 | | DEC | 258 | |
| 2 | MW 8 | | DEC | 1 | |
| 3 | DB1.DBW 0 | | HEX | W#16#0064 | |
| 4 | DB1.DBB 2 | | BIN | 2#0000_0011 | |
| 5 | DB1.DBB 3 | | BIN | 2#0000_0000 | |
| 6 | MW 10 | | HEX | W#16#0294 | |
| 7 | M 2.0 | | BOOL | true | true |
| 8 | DB1.DBB 6 | | BIN | 2#0000_0011 | |

SFC51\SIMATIC 300(1)\...\S7 程序(1)　　RUN　　Abs <

图 12-55　变量表监控

的 IO 设备的状态，1 表示 IO 设备存在；0 表示 IO 设备丢站。其他的高字节中的位（Bit）的含义依次类推，表示对应 IO 设备的状态。

4）对数据块 DB1 进行监控，如图 12-56 所示，同样可以解读出设备的通信状态。

**3. 用 FC10 进行故障诊断**

（1）FC10（AG_CNTRL）的功能

| 地址 | 名称 | 类型 | 初始值 | 实际值 |
|---|---|---|---|---|
| 0.0 | A[0] | BYTE | B#16#0 | B#16#0D |
| 1.0 | A[1] | BYTE | B#16#0 | B#16#15 |
| 2.0 | A[2] | BYTE | B#16#0 | B#16#00 |
| 3.0 | A[3] | BYTE | B#16#0 | B#16#00 |
| 4.0 | A[4] | BYTE | B#16#0 | B#16#03 |
| 5.0 | A[5] | BYTE | B#16#0 | B#16#E8 |
| 6.0 | A[6] | BYTE | B#16#0 | B#16#03 |
| 7.0 | A[7] | BYTE | B#16#0 | B#16#0F |
| 8.0 | A[8] | BYTE | B#16#0 | B#16#10 |
| 9.0 | A[9] | BYTE | B#16#0 | B#16#10 |
| 10.0 | A[10] | BYTE | B#16#0 | B#16#10 |
| 11.0 | A[11] | BYTE | B#16#0 | B#16#10 |
| 12.0 | A[12] | BYTE | B#16#0 | B#16#00 |
| 13.0 | A[13] | BYTE | B#16#0 | B#16#00 |
| 14.0 | A[14] | BYTE | B#16#0 | B#16#00 |
| 15.0 | A[15] | BYTE | B#16#0 | B#16#00 |
| 16.0 | A[16] | BYTE | B#16#0 | B#16#00 |
| 17.0 | A[17] | BYTE | B#16#0 | B#16#00 |
| 18.0 | A[18] | BYTE | B#16#0 | B#16#00 |
| 19.0 | A[19] | BYTE | B#16#0 | B#16#00 |

图 12-56　数据块 DB1 监控

在故障诊断时，可以使用 FC10（AG_CNTRL）程序块对连接进行诊断。具体如下：

1）读出连接信息。根据状态信息，可以确定所有或单个通信模块 CP 连接是否有用。

2）复位已组态的连接。可以复位个别或所有通信模块 CP 的连接。终止活动连接，并重新建立连接。

（2）FC10（AG_CNTRL）参数简介

FC10（AG_CNTRL）的指令和参数见表 12-15。

**表 12-15　FC10（AG_CNTRL）的指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存储区 |
|---|---|---|---|---|
| “AG_CNTRL”<br>EN　ENO<br>ACT　DONE<br>ID　ERROR<br>LADDR　STATUS<br>CMD　RESULT1<br>RESULT2 | EN | BOOL | 使能输入端 | |
| | ACT | BOOL | ACT＝1 时激活处理 | I、Q、M、D、L |
| | ID | INT | 指定连接的数目。可以在组态中找到连接数目 | I、Q、M、D、L 、常数 |
| | LADDR | WORD | 模块起始地址，在组态时，可看到 | I、Q、M、D、L |
| | CMD | INT | 详细参见下文 | I、Q、M、D、L |
| | DONE | BOOL | 为 1 时，完成作业 | I、Q、M、D、L |
| | ERROR | BOOL | 为 1 时，有错误 | I、Q、M、D、L |
| | STATUS | WORD | 状态代码 | I、Q、M、D、L |
| | RESULT1 | DWORD | 根据发送至 FC AG_CNTRL 的命令返回的信息 | I、Q、M、D、L |
| | RESULT2 | DWORD | 根据发送至 FC AG_CNTRL 的命令返回的信息（第二部分，仅对 S7－400） | I、Q、M、D、L |

（3） FC10 （AG_CNTRL） 的应用举例

**【例 12-6】** 某控制系统采用工业以太网通信，其网络组态如图 12-57 所示，请用 FC10 指令读取通信反馈信息。

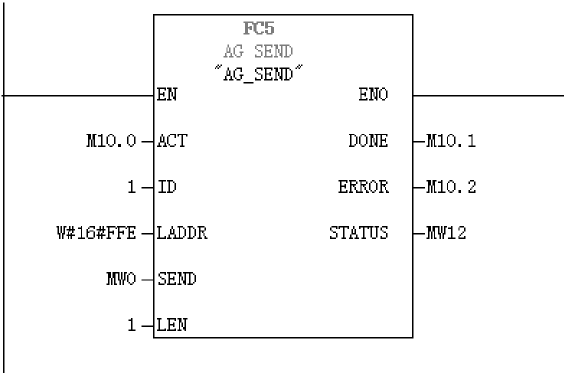
**解：** 硬件配置和网络组态不再赘述，前面章节已经讲解。

图 12-57　网络组态图

1） 客户端是 S7-400，客户端的程序如图 12-58 所示。

2） 编写服务器端梯形图程序，如图 12-59 所示。

⊟ **程序段 1**：标题：




⊟ **程序段 2**：标题：

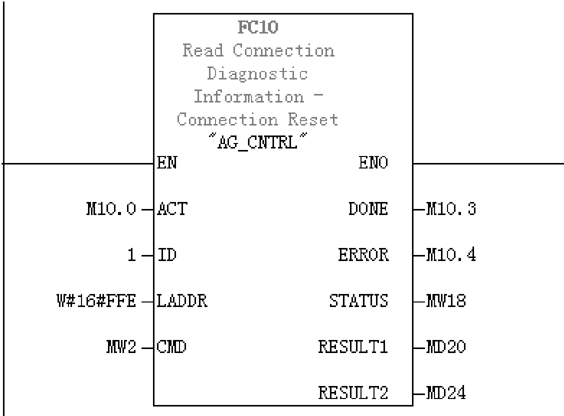



图 12-58　客户端梯形图

⊟ **程序段 1**：标题：

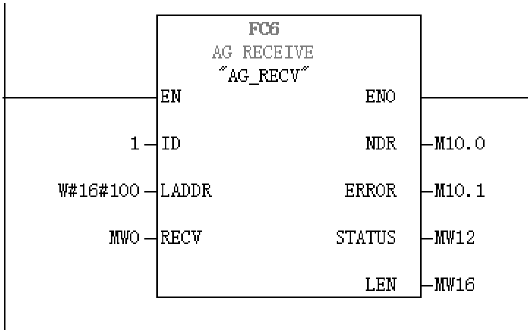



图 12-59　服务器端梯形图

3） 在客户端的“块”中创建变量表；在变量表中监控反馈信息。

① 使 M10.0=1，激活通信状态，使 MW2（CMD）=16#0001。反馈信息读取到 MD20 中，如图 12-60 所示。当 CMD=1 时，反馈值的低 16 位有效，其各位含义见表 12-16。




图 12-60　用变量表监控

**表 12-16　CMD =1 时，反馈值的低 16 位含义**

| CMD 值 | 位　　号 | | 含　　义 |
|---|---|---|---|
| 1 | 位 0 ~3 用于发送方向的代码 | 位 0 | 连接类型。0：无发送和接收连接；1：为发送和接收作业保留的连接 |
| | | 位 1 | 当前作业的状态。0：当前没在执行发送作业；1：正在执行发送作业 |
| | | 位 2、3 | 上次作业。00：没有关于上次发送作业的信息；01：上次发送作业成功完成；10：上次发送作业未成功完成 |
| | 位 4 ~7 用于接收方向的代码 | 位 4 | 连接类型。0：无发送和接收连接；1：为发送和接收作业保留的连接 |
| | | 位 5 | 当前作业的状态。0：当前没在执行接收作业；1：正在执行接收作业 |
| | | 位 6、7 | 上次作业。00：没有关于上次接收作业的信息；01：上次接收作业成功完成；10：上次接收作业未成功完成 |
| | 位 8 ~11 用于 FETCH/WRITE 的代码 | 位 8 | 连接类型。0：无 FETCH 连接；1：为 FETCH 作业保留的连接 |
| | | 位 9 | 连接类型。0：无 WRITE 连接；1：为 WRITE 作业保留的连接 |
| | | 位 10 | 作业状态（FETCH/WRITE）。0：作业状态正常；1：作业状态不正常 |
| | | 位 11 | FETCH/WRITE 作业的状态。0：无激活的作业；1：来自 LAN 的作业激活 |
| | 位 12 ~15 常规 CP 信息 | 位 12、13 | 连接状态的信息。00：连接被终止；01：连接建立激活；10：连接终止激活；11：连接已经建立 |
| | | 位 14 | CP 信息。0：CP 处于 STOP 状态；1：CP 处于 RUN 状态 |
| | | 位 15 | 复位 ID。0：FC10 尚未复位连接或复位 ID 已清除；1：控制块执行了一个连接复位 |

如图 12-60 所示，反馈信息为 2#0000_0000_0000_0000_0111_0000_0001_0111。以下根据表 12-4 解释其含义。

- 位 0 是 1，表示为发送和接收作业保留的连接。
- 位 1 是 1，表示正在执行发送作业。
- 位 2、3 是 01，表示上次发送作业成功完成。
- 位 4 是 0，表示无接收连接。
- 位 5 是 0，表示没有执行接收作业。
- 位 6、7 是 01，表示上次接收作业成功完成。
- 位 12、13 是 11，表示处于连接状态。
- 位 14 是 1，表示 CP 处于 RUN 状态。

● 位 15 是 0，表示 FC10 尚未复位连接或复位 ID 已清除。

综合以上信息可知：以太网处于连接状态，CP 运行正常，发送数据正常。

② 使 M10.0 = 1，激活通信状态，使 MW2（CMD）= 16#0002。反馈信息读取到 MD20 中，如图 12-61 所示。当 CMD = 2 时，反馈值的低 16 位有效。

当 CMD = 16#0000_0001 时，表示成功地将复位作业传送给 CP。触发连接中止及后续的连接建立。

当 CMD = 16#0000_0002 时，表示无法将复位作业传送给 CP，原因是 CP 上的服务尚未起动（例如，CP 处于 STOP 状态）。

如图 12-53 所示，反馈信息为 2#0000_0000_0000_0000_0000_0000_0000_0001 = 16#0000_0001。表示成功地将复位作业传送给 CP。

③ 使 M10.0 = 1，激活通信状态，使 MW2（CMD）= 16#0003，获取所有的连接状态信息，两个双字反馈值有效。反馈信息读取到 MD20 和 MD24 中，如图 12-62 所示。当为 1 时，表示连接，为 0 时表示连接终止或者没有组态。图 12-62 中反馈信息为 2#0000_0000_0000_0000_0000_0000_0000_0000，表示都未连接。如果为反馈信息为 2#0000_0000_0000_0000_0000_0000_0000_0111，则表示 ID 号为 1、2 和 3 的处于连接状态。

图 12-61　用变量表监控

图 12-62　用变量表监控

**4. 用 FB10 进行故障诊断**

使用 AG_CNTEX（FB10）程序块可以对连接进行诊断，并可使用 Ping 命令通过网络对设备进行寻址。必要时，可以使用 AG_CNTEX 重新初始化连接建立过程。

（1）FB10 的功能

AG_CNTRL（FC10）和与程序块 AG_CNTEX（FB10）相比，程序块 AG_CNTEX 可提供更多功能。AG_CNTEX 包含了 AG_CNTRL 的所有功能，可以在用户程序中以相同方式使用这些功能。其具体功能如下：

1）读出连接信息。根据状态信息，可以确定复位所有或单个 CP 连接是否有用。

2）复位已组态的连接。可以复位 CP 的个别或所有连接。

3）终止活动连接，并重新建立连接。

4）读出 CP 中组态的连接类型（相对于 AG_CNTRL 的扩展功能）。

5）发送 PING 命令（相对于 AG_CNTRL 的扩展功能）。

6）可以检查网络中的特定节点是否可到达。

（2）FB10（AG_CNTEX）参数简介

FB10（AG_CNTEX）的指令参数含义见表 12-17。

**表 12-17　FB10（AG_CNTEX）的指令和参数**

| LAD | 参数 | 数据类型 | 说　　明 | 存储区 |
|---|---|---|---|---|
| “AG_GNTEX”<br>EN　ENO<br>ACT　DONE<br>ID　ERROR<br>LADDR　STATUS<br>CMD　RESULT1<br>PING　RESULT2 | EN | BOOL | 使能输入端 | I、Q、M、D、L |
| | ACT | BOOL | ACT = 1 时激活处理 | I、Q、M、D、L |
| | ID | INT | 指定连接的数目。可以在组态中找到连接数目 | I、Q、M、D、L 、常数 |
| | LADDR | WORD | 模块起始地址，在组态时，可看到 | I、Q、M、D、L |
| | CMD | INT | 发给 AG_CNTEX 的命令 | I、Q、M、D、L |
| | PING | ANY | 引用包含 Ping 命令数据结构的数据块，详细参见下文 | |
| | DONE | BOOL | 为 1 时，完成作业 | I、Q、M、D、L |
| | ERROR | BOOL | 为 1 时，有错误 | I、Q、M、D、L |
| | STATUS | WORD | 状态代码 | I、Q、M、D、L |
| | RESULT1 | DWORD | 根据发送至 AG_CNTEX 的命令返回的信息 | I、Q、M、D、L |
| | RESULT2 | DWORD | 根据发送至 AG_CNTEX 的命令返回的信息 | I、Q、M、D、L |

PING 参数引用数据块，其数据结构见表 12-18。

**表 12-18　PING 参数引用数据块的数据结构**

| 参　　数 | 数 据 类 型 | 取 值 范 围 | 说　　明 |
|---|---|---|---|
| IP 地址 | ARRAY[1..4] of Byte | | 存放 IP |
| TIMEOUT | INT | 1..60000 ms | 可作为选项指定；默认值 = 1000 ms |
| SIZE | INT | 1..1000 B | 可作为选项指定；默认值 = 32 B |

（3）FB10 指令的应用举例

**【例 12-7】**某控制系统采用工业以太网通信，其网络组态如图 12-57 所示，请用 FB10 指令读取通信反馈信息，并用 PING 功能检查网络是否正常连接。

**解：**FB10 指令读取通信反馈信息方法与 FC10 读取反馈信息的方法完全相同，在此不详细介绍，仅将反馈信息列出。

1）创建自定义数据类型 UDT1，如图 12-63 所示，其符号名为“PingA”。

| 地址 | 名称 | 类型 | 初始值 | 注释 |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | Ping | ARRAY[0..3] | | |
| *1.0 | | BYTE | | |
| +4.0 | TimeOut | INT | 0 | |
| +6.0 | Size | INT | 0 | |
| =8.0 | | END_STRUCT | | |

图 12-63　创建 UDT1

2）创建数据块 DB1，其符号名为“Ping”，创建 3 个变量，分别是 Ping1、Ping2 和 Ping3，其数据类型为前面创建的 UDT1，即“PingA”，如图 12-64 所示。单击程序编辑器菜单中的“视图”→“数据视图”，弹出如图 12-65 所示的界面。

注意：不创建 UDT1 也可行，但在创建数据块时就比较麻烦，如果以太网连接的站点越多，使用自定义数据类型就越方便。

在图 12-65 中，有三个 IP 地址分别是 192.168.0.3（192 的十六进制是 C0，存储在 Ping1.Ping[0]中；168 的十六进制是 A8，存储在 Ping1.Ping[1]中）、192.168.0.4 和 192.168.0.5。

| 地址 | 名称 | 类型 | 初始值 | 注释 |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | Ping1 | "PingA" | | |
| +8.0 | Ping2 | "PingA" | | |
| +16.0 | Ping3 | "PingA" | | |
| =24.0 | | END_STRUCT | | |

图 12-64　创建 DB1（声明视图）

| 地址 | 名称 | 类型 | 初始值 | 实际值 | 注释 |
|---|---|---|---|---|---|
| 0.0 | Ping1.Ping[0] | BYTE | B#16#0 | B#16#C0 | |
| 1.0 | Ping1.Ping[1] | BYTE | B#16#0 | B#16#A8 | |
| 2.0 | Ping1.Ping[2] | BYTE | B#16#0 | B#16#0 | |
| 3.0 | Ping1.Ping[3] | BYTE | B#16#0 | B#16#3 | |
| 4.0 | Ping1.TimeOut | INT | 0 | 1000 | |
| 6.0 | Ping1.Size | INT | 0 | 32 | |
| 8.0 | Ping2.Ping[0] | BYTE | B#16#0 | B#16#C0 | |
| 9.0 | Ping2.Ping[1] | BYTE | B#16#0 | B#16#A8 | |
| 10.0 | Ping2.Ping[2] | BYTE | B#16#0 | B#16#0 | |
| 11.0 | Ping2.Ping[3] | BYTE | B#16#0 | B#16#4 | |
| 12.0 | Ping2.TimeOut | INT | 0 | 1000 | |
| 14.0 | Ping2.Size | INT | 0 | 32 | |
| 16.0 | Ping3.Ping[0] | BYTE | B#16#0 | B#16#C0 | |
| 17.0 | Ping3.Ping[1] | BYTE | B#16#0 | B#16#A8 | |
| 18.0 | Ping3.Ping[2] | BYTE | B#16#0 | B#16#0 | |
| 19.0 | Ping3.Ping[3] | BYTE | B#16#0 | B#16#5 | |
| 20.0 | Ping3.TimeOut | INT | 0 | 1000 | |
| 22.0 | Ping3.Size | INT | 0 | 32 | |

图 12-65　DB1 的数据视图

3）客户端是 S7-400 PLC，客户端的程序如图 12-66 所示。

4）编写服务器端梯形图程序，如图 12-67 所示。

5）在客户端的“块”中，创建变量表，在变量表中，监控反馈信息。

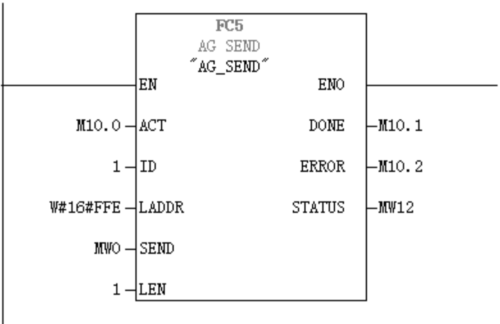
**5. 用 FC3 对 CP342-5 的从站进行故障诊断**

（1）FC3（DP_DIAG）的功能

程序块 FC3（DP_DIAG）用于请求诊断信息，通过指定一个站地址，还可以为一个指定的从站请求诊断数据。为了将诊断数据传送到 CPU，应该在 CPU 中保留一个内存区域，然后在调用中指定该区域。该内存区域可以是一个数据块区或一个位存储区。还可以在作业中指定可用内存区域的最大长度。

FC3（DP_DIAG）只有在 DP 主站模式中才起作用，从站模式无效。其具体功能如下：

⊟ **程序段 1**：标题：




⊟ **程序段 2**：标题：

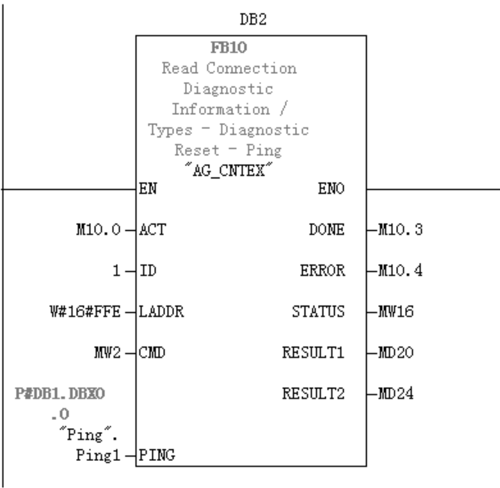



图 12-66　客户端梯形图

⊟ **程序段 1**：标题：

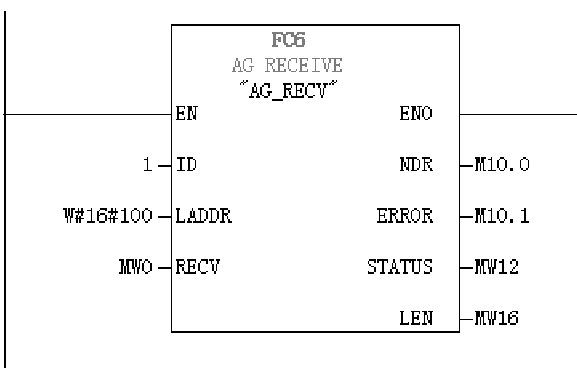



图 12-67　服务器端梯形图

1）请求 DP 站列表。

2）请求 DP 诊断列表。

3）请求 DP 单个状态。

4）非周期性地读取 DP 从站的输入/输出数据。

5）读取较早的 DP 单个诊断信息。

6）读取 DP 状态。

7）读取用于 PLC/CP 停止的 DP 模式。

8）读取 DP 从站的当前状态。

（2）FC3（DP_DIAG）指令参数

FC3（DP_DIAG）的指令参数含义见表 12-19。

**表 12-19　FC3（DP_DIAG）的指令和参数**

| LAD | 参　数 | 数据类型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| | EN | BOOL | 使能输入端 | I、Q、M、D、L |
| | CPLADDR | WORD | CP 模块起始地址 | I、Q、M、D、L |

（续）

| LAD | 参　数 | 数据类型 | 说　明 | 存　储　区 |
|---|---|---|---|---|
| "DP_DIAG"<br>EN　ENO<br>CPLADDR　NDR<br>DTYPE　ERROR<br>STATION　STATUS<br>DIAG　DIAGLNG | DTYPE | BYTE | 诊断类型。0：站列表；1：诊断列表；2：单个站当前诊断信息；3：原诊断信息；4：读取 DP 状态；5：读取 CPU STOP 的状态；6：读取 CP STOP 的状态；7：读取输入数据（非周期地）；8：读取输出数据（非周期地；10：读取 DP 从站的当前状态 | I、Q、M、D、L、常数 |
| | STATION | BYTE | DP 从站的站地址 | I、Q、M、D、L |
| | DIAG | ANY | 指定数据区地址和长度 | I、Q、M、D、L |
| | NDR | ANY | 该参数指示是否已接受新数据 | |
| | DONE | BOOL | 为 1 时，完成作业 | I、Q、M、D、L |
| | ERROR | BOOL | 为 1 时，有错误 | I、Q、M、D、L |
| | STATUS | WORD | 状态代码 | I、Q、M、D、L |
| | DIAGLNG | WORD | CP 可以使用的数据的实际长度（单位为字节） | I、Q、M、D、L |

（3）FC3 指令的应用举例

**【例 12-8】**某控制系统采用 PROFIBUS－DP 通信，其硬件组态如图 12-68 所示，请用 FC3 指令读取通信诊断信息。

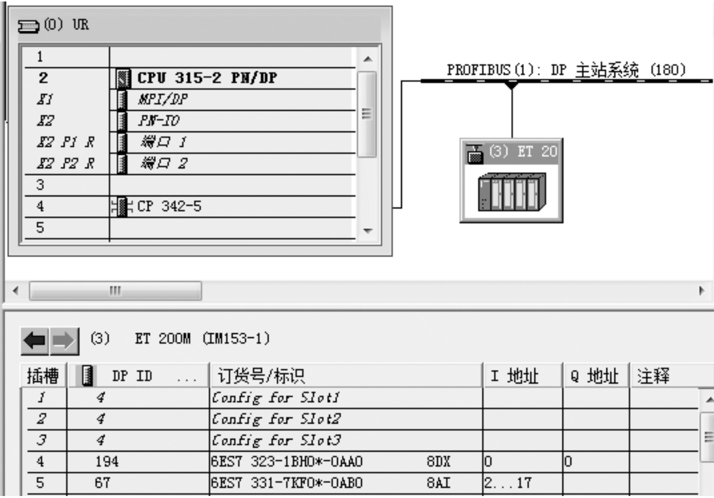



图 12-68　硬件组态图

**解：**

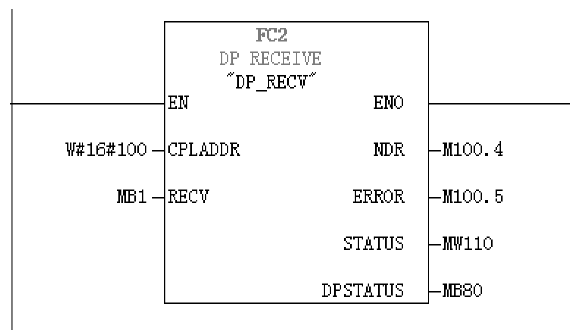
1）创建一个项目，硬件组态如图 12-68 所示，具体操作步骤参考 8.4.6（CP342－5 PROFIBUS－DP 作主站的通信）。

2）创建一个数组。在创建数组前，先创建共享数据块 DB1，再在数据块中创建一个数组（包含 16 个元素，数据类型为字节），如图 12-69 所示。

| 地址 | 名称 | 类型 | 初始值 | 注释 |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | AA | ARRAY[0..15] | | 16个字节元素的数组 |
| *1.0 | | BYTE | | |
| =16.0 | | END_STRUCT | | |

图 12-69　数组

3）编写梯形图程序如图 12-70 所示。



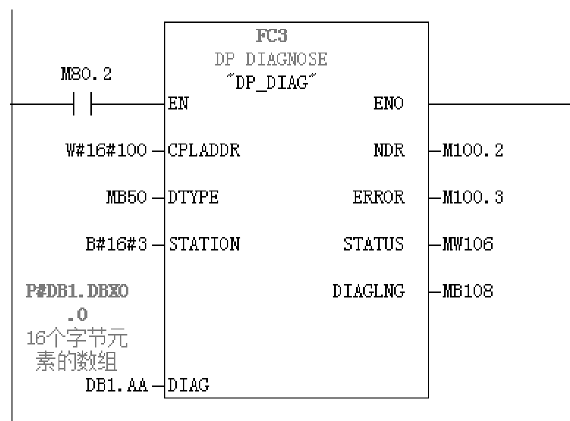




图 12-70　梯形图

M80.2 为 0 表示不存在新的诊断数据，当 M80.2 为 1 表示有新的诊断数据。

① 当 DTYPE(MB50) = 0 时，表示读取的数据为 DP 站列表，站地址的范围是 0 ~ 127。

DP 站的列表的地址用 FC3 的输入参数 DIAG 设置，长度为 16 B（128 位），每一位对应一个 DP 地址，即对应于一个 DP 从站，如图 12-71 所示。

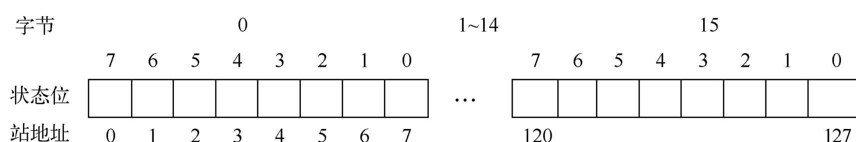



图 12-71　DP 站列表中从站状态位

状态位为 0 的含义有三种情况：从站处于周期性传输状态、没有使用该站以及该从站输入/输出数据的长度组态为 0。

状态位为 1 的含义是从站没有处于周期性传输状态。

当 DTYPE(MB50)=0 时，数据块中的数组 AA 中存储的数据就是位状态，使数组处于监控状态，如图 12-72 所示。数字 AA 的第一个字节 AA[0]=16#10=2#0001_0000，对应的就是 3 号站，其含义是 3 号站没有处于周期性传输状态。可能的原因是 3 号站掉电或者电缆断开等原因。假如，AA[0]=16#18=2#0001_1000，表示的含义是 3、4 号站没有处于周期性传输状态。

| 地址 | 名称 | 类型 | 初始值 | 实际值 | 注释 |
|---|---|---|---|---|---|
| 0.0 | AA[0] | BYTE | B#16#0 | B#16#10 | 16个字节元素的数组 |
| 1.0 | AA[1] | BYTE | B#16#0 | B#16#00 | |
| 2.0 | AA[2] | BYTE | B#16#0 | B#16#00 | |
| 3.0 | AA[3] | BYTE | B#16#0 | B#16#00 | |
| 4.0 | AA[4] | BYTE | B#16#0 | B#16#00 | |
| 5.0 | AA[5] | BYTE | B#16#0 | B#16#00 | |
| 6.0 | AA[6] | BYTE | B#16#0 | B#16#00 | |
| 7.0 | AA[7] | BYTE | B#16#0 | B#16#00 | |
| 8.0 | AA[8] | BYTE | B#16#0 | B#16#00 | |
| 9.0 | AA[9] | BYTE | B#16#0 | B#16#00 | |
| 10.0 | AA[10] | BYTE | B#16#0 | B#16#00 | |
| 11.0 | AA[11] | BYTE | B#16#0 | B#16#00 | |
| 12.0 | AA[12] | BYTE | B#16#0 | B#16#00 | |
| 13.0 | AA[13] | BYTE | B#16#0 | B#16#00 | |
| 14.0 | AA[14] | BYTE | B#16#0 | B#16#00 | |
| 15.0 | AA[15] | BYTE | B#16#0 | B#16#00 | |

图 12-72　数组 AA

② 当 DTYPE(MB50)=2 时，表示读取的单个站的当前诊断信息。使数组 AA 处于监控状态，如图 12-73 所示。

前 3 个字节(AA[0]~AA[2])表示站的状态；第 4 个字节表示主站地址，本例为 16#02；第 5、6 个字节表示的是制造商标识号；第 7 个字是固定数值；从第 9 个字开始表示的是诊断信息。

| 地址 | 名称 | 类型 | 初始值 | 实际值 | 注释 |
|---|---|---|---|---|---|
| 0.0 | AA[0] | BYTE | B#16#0 | B#16#00 | 16个字节元素的数组 |
| 1.0 | AA[1] | BYTE | B#16#0 | B#16#0C | |
| 2.0 | AA[2] | BYTE | B#16#0 | B#16#11 | |
| 3.0 | AA[3] | BYTE | B#16#0 | B#16#02 | |
| 4.0 | AA[4] | BYTE | B#16#0 | B#16#80 | |
| 5.0 | AA[5] | BYTE | B#16#0 | B#16#1D | |
| 6.0 | AA[6] | BYTE | B#16#0 | B#16#43 | |
| 7.0 | AA[7] | BYTE | B#16#0 | B#16#00 | |
| 8.0 | AA[8] | BYTE | B#16#0 | B#16#00 | |
| 9.0 | AA[9] | BYTE | B#16#0 | B#16#00 | |
| 10.0 | AA[10] | BYTE | B#16#0 | B#16#00 | |
| 11.0 | AA[11] | BYTE | B#16#0 | B#16#00 | |
| 12.0 | AA[12] | BYTE | B#16#0 | B#16#00 | |
| 13.0 | AA[13] | BYTE | B#16#0 | B#16#00 | |
| 14.0 | AA[14] | BYTE | B#16#0 | B#16#00 | |
| 15.0 | AA[15] | BYTE | B#16#0 | B#16#00 | |

图 12-73　数组 AA

当然，DTYPE 还有其他的取值，请读者自己参考相关手册。

此外，还有一些其他的诊断方法，如使用 FB126 和 FC125 诊断，请读者自行参考相关资料。

### 12.2.6　用 BT200 PROFIBUS 总线物理测试工具诊断故障

#### 1. BT200 的功能

利用 RS-485 电气总线电缆，BT200 提供用于 PROFIBUS-DP 系统的诊断选项，而且

无需额外的测量工具。BT200 外形如图 12-74 所示。

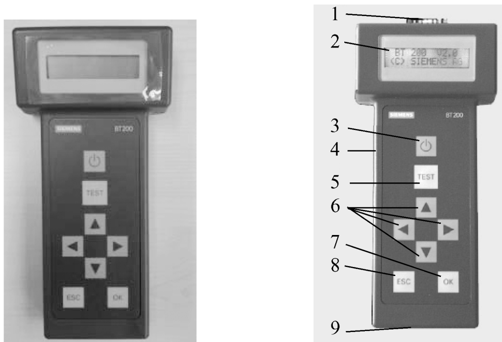



图 12-74　BT200 外形

1—PROFIBUS-DP 连接口　2—显示界面　3—开关机按钮　4—充电电缆连接口
5—测试按钮（开始测）　6—箭头指示按钮　7—确认按钮　8—退出按钮　9—充电触点

BT200 结构紧凑、容易操作，可供安装人员、调试人员和维修工程师使用。具体功能如下：

（1）测试 PROFIBUS 电缆

1）电缆断线/屏蔽故障。

2）电缆之间或者电缆和屏蔽层之间短路。

3）在数据电缆断线或短路情况下的故障定位。

4）导致故障的反射的检测。

5）电缆更换。

6）已安装电缆的长度。

（2）检查从站的可访问性

1）可访问从站的列表。

2）单个从站的特定寻址。

3）检查主站和从站的 RS-485 接口。

4）显示 PROFIBUS-DP 的站地址。

**2. BT200 的使用**

（1）总体说明

通常情况下，BT200 通过 ON/OFF 键开关机，开机时必须长时间按下此键，直到显示屏上出现响应。以下是几种常见的显示状态：

1）待机：开机后此画面大约显示 2 s 左右，如图 12-75 所示。

2）电池：待机后，电池充电指示器显示大约 2 s，如图 12-76 所示。

3）起动状态：电池充电指示器关闭后，BT200 就切换到正常模式，并且显示布线测试的开始窗口，如图 12-77 所示。




图 12-75　BT200 待机显示




图 12-76　BT200 电池显示

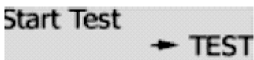



图 12-77　BT200 起动

（2）普通线路测试

测试是在 BT200 和测试插头之间的总线网段完成的，如图 12-78 所示。在安装阶段，可以测试插头之间的路径测试插头始终连接到总线网段的一端。此时还可以检测到测试路径外部的短路。总线网段的起始端和末端必须连接一个终端电阻。

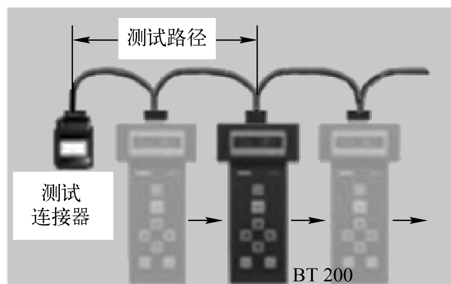



图 12-78　用 BT200 测试线路

（3）测试步骤

1）确认有站点连接到总线电缆。

2）测试插头必须连接到要测试的总线电缆的一端。

3）测试插头末端上的终端电阻或者 BT200 的连接点上的终端电阻至少有一端激活。

4）按下“Test”按钮来起动测试。

5）如果测试正常进行，会显示如图 12-79（表示 1 端安装了终端电阻，安装没有完成）和图 12-72（表示 2 端安装了终端电阻，安装完成）两个界面中的一个。

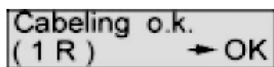



图 12-79　BT200 测试显示 - 1R

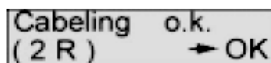



图 12-80　用 BT200 测试显示 - 2R

（4）测试过程中的错误显示

1）站点测试。需要检查是否所有的连接器都已经从站点上移开（不能带电测量），或者检查所有站点是否都已断电，如图 12-81 所示。

2）交换线芯。在相应的连接器中交换线芯，交换 RS - 485 的 A、B 线芯，即绿线和红线交换接线孔，如图 12-82 所示。

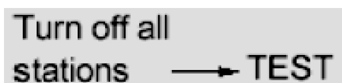



图 12-81　BT200 测试显示 - 关闭电源

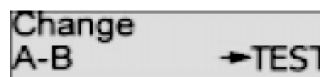
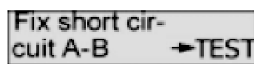



图 12-82　BT200 测试显示 - 交换线芯

3）短路。总线的短路分为三种情况：RS - 485 的 A、B 线短路、RS - 485 的 A 线与屏蔽层短路和 RS - 485 的 B 线与屏蔽层短路，如图 12-83 所示。




a)




b)




c)

图 12-83　BT200 测试显示 - 短路

4）没有接入终端电阻或接入超过两个终端电阻。在总线网段的起始端或末端分别接入一个终端电阻（至少一端被激活），除这两个终端电阻之外，必须移除或禁止所有总线网段上的其他终端电阻。

图 12-84a、b 分别显示接通末端的终端电阻和接通两端的终端电阻。

图 12-85a、b 分别显示移除末端的终端电阻和移除两端的终端电阻。

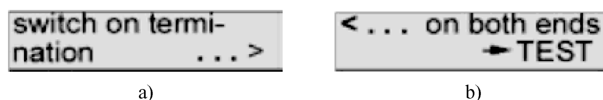



图 12-84　BT200 测试显示 – 接入电阻

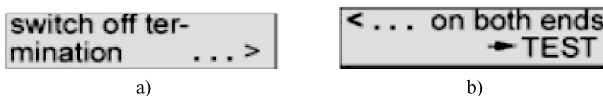



图 12-85　BT200 测试显示 – 移除电阻

5）断线。如果多根芯线或至少一根芯线与屏蔽层断开或未连接，那么 BT200 不能确定故障的确切位置。

如图 12-86 所示，显示了断线的 4 种情况，分别是芯线和屏蔽层断开、测试器未连接好（图 12-78a）、A 芯线断开（图 12-86b）、B 芯线和断开（图 12-86c）以及屏蔽层断开（图 12-6d）。

**3. 测试连线示意图**

BT200 测试连线的示意图如图 12-87 所示。

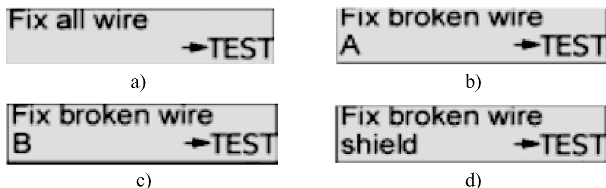



图 12-86　BT200 测试显示 – 断线

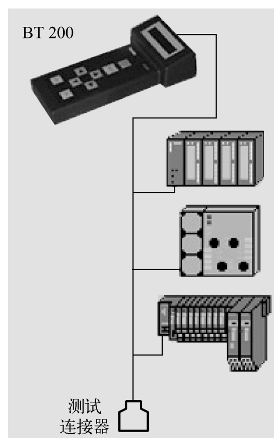



图 12-87　BT200 测试连线示意图

## 12.2.7　用 PC 网卡/CP443 – 1/CP343 – 1 等通信处理器诊断以太网故障

用 CP343 – 1、CP443 – 1 和 CP1613A2 等通信处理器（甚至 PC 的网卡）进行以太网的故障诊断简单易行，在通信的故障诊断中较为常用，但应用的前提是系统中安装有以上通信处理器的一种或者几种。以下分别介绍。

**1. 用 PC 网卡（或者 CP1613 A2）诊断以太网故障**

用 PC 网卡（或者 CP1613 A2）可以非常便捷地检测网络是否已经联通，具体使用 "PING" 功能进行测试。方法如下：

1）单击 Windows 系统的“开始”按钮，在 Windows 窗口的左下方弹出一个方框，输入“PING + IP 地址”，这里的 IP 地址就是试图联通目标的 IP 地址（例如本例是

192.168.0.2），如图 12-88 所示，单击计算机的“Enter”键，这时计算机试图和 IP 地址为 192.168.0.2 的对象连接。

2）PING 运行结果解读。如图 12-89 所示，表示计算机的 IP 地址是“192.168.0.188”，而目标地址是“192.168.0.2”，无法建立连接。无法建立连接原因大致有如下几种：




图 12-88　PING 目标对象

① IP 地址设置不正确，例如 PC 的 IP 地址与目标地址不在同一网段，本例中计算机的 IP 地址是“192.168.0.188”，而目标地址是“192.168.0.2”，显然在一个网段。

② 子网掩码不相同。应设置相同，一般设为“255.255.255.0”。

③ 网线断开。如网线断裂，网络接头松动，特别是 RJ45 接头容易松动等。

④ 试图连接的目标站点断电。

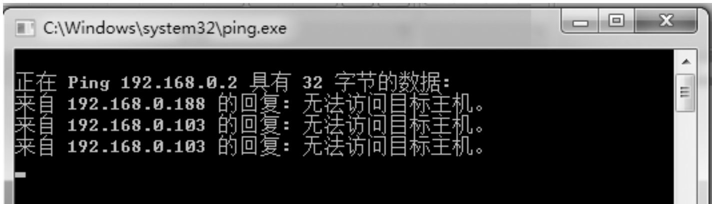



图 12-89　PING 运行结果－未联通

检查发现，目标地址是“192.168.0.2”的 CP343－1 模块断电，通电后，再做“PING”操作，如图 12-90 所示。表明：目标地址是“192.168.0.2”的 CP343－1 模块，在 1 ms 内做出响应，说明 CP343－1 模块到交换机是联通的。

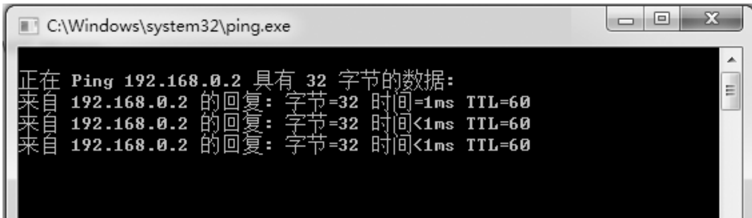



图 12-90　PING 运行结果－联通

**2. 用 CP443－1（CP343－1）诊断以太网故障**

CP443－1（CP343－1）诊断以太网连接状态也比较简单，以下用一个例子介绍。

**【例 12-9】**某控制系统采用工业以太网通信，其网络组态如图 12-91 所示，请用 CP443－1 诊断网络的连接状态。

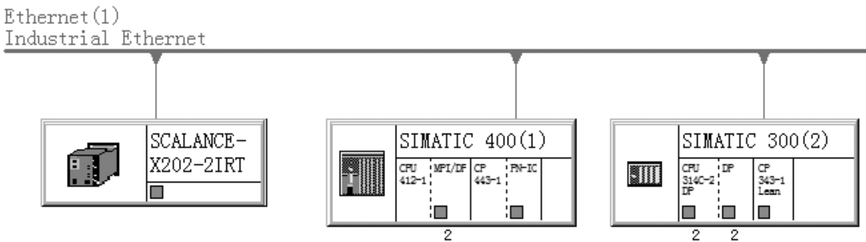



图 12-91　网络组态

**解**：本例客户端的 IP 地址为 192.168.0.1，服务器端的 IP 地址为 192.168.0.3。具体方法如下：

**方法 1**：

1）打开硬件组态界面，选中“CP443-1”（或者 CP343-1），单击鼠标右键，单击弹出的快捷菜单中的“对象属性”子菜单，弹出如图 12-92 所示界面，选中“诊断”选项卡，单击“运行”按钮，弹出诊断界面。




图 12-92　打开诊断

2）如图 12-93 所示，单击“监控”按钮，单击目录树“连接”→“TCP”→“TCP 连接 1”，可以看到界面的右侧显示：连接已建立，建立连接的伙伴地址为 192.168.0.3。说明以太网连接没有断开。

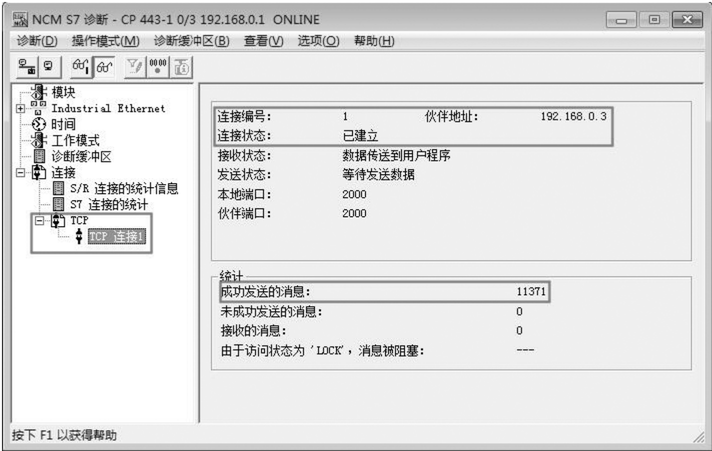



图 12-93　连接状态

3）如图 12-94 所示，单击“监控”按钮，单击目录树“连接”→“诊断缓冲区”，




图 12-94　诊断状态

可以看到界面的右侧显示，连接已经由连接伙伴终止，说明以太网连接断开了，进一步检查，发现 IP 地址 192. 168. 0. 3 伙伴站断电了，至此故障原因找到了。

**方法 2：**

单击“选项”→“Ping”，如图 12-95 所示。弹出如图 12-96 所示的界面，单击“开始”按钮，可以看到在 100 ms 内，IP 地址为 192. 168. 0. 3 的目标对象做出响应，可见以太网处于连接状态。




图 12-95　打开 Ping 界面




图 12-96　Ping 的结果

### 12. 2. 8　用 CP5611/CP5621/CP342 - 5 等通信处理器诊断通信故障

用 CP5611、CP5613、CP5614、CP5621、CP5512、CP342 - 5 和 CP443 - 5 等通信处理器进行 PROFIBUS 故障诊断，简单易行，在通信的故障诊断中较为常用，但应用的前提是系统中安装有以上通信处理器的一种或者几种。

CP5611、CP5613 和 CP5621 通信处理器安装在台式的主板的 PCI 接口插槽（也就是插装显卡、网卡的插槽）中，即插即用，不需要安装驱动程序。CP5511 和 CP5512 用于笔记本电脑。CP342 - 5 和 CP443 - 5 是 S7 - 300 和 S7 - 400 的 PROFIBUS 的通信模块。以下分别介绍。

**1. 用 CP5611 通信处理器诊断故障**

CP5611、CP5613、CP5621、CP5511 和 CP5512 通信处理器，不仅能诊断 PROFIBUS 通信的故障，而且能诊断 MPI 的通信故障。以下仅以 CP5611 为例，讲解诊断 PROFIBUS 通信的故障。

**【例 12-10】** 某控制系统采用 PROFIBUS 通信，其硬件组态如图 12 - 97 所示，请用 CP5611 诊断网络的连接状态。

**解：**

（1）设置 PG/PC 接口。

在 SIMATIC Manager 界面中，单击“选项”→“设置 PG/PC 接口”，弹出如图 12-98 所示的界面。并把“CP5611（PROFIBUS）”激活，单击“诊断”按钮。

打开“设置 PG/PC 接口”界面，还有另外一种方法，即直接打开控制面板，双击“设置 PG/PC 接口”，如图 12-99 所示（如果读者的控制面板中无“设置 PG/PC 接口”，原因

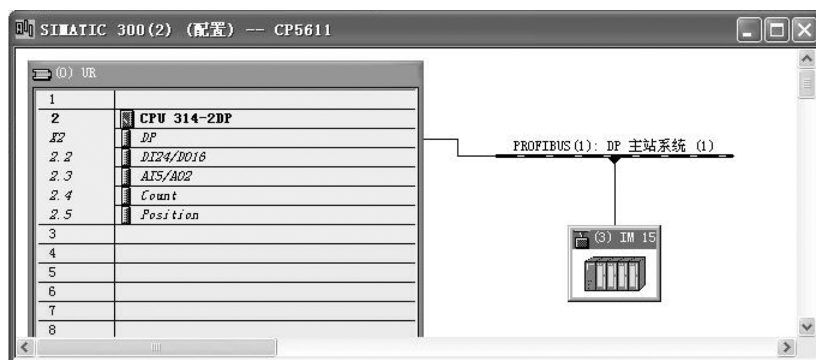



图 12-97　硬件组态




图 12-98　设置 PG/PC 接口

是读者的计算机没有安装 STEP 7 等软件)。




图 12-99　打开“设置 PG/PC 接口”

（2）诊断

如图12-100所示，单击“测试”按钮，“测试”按钮的右侧出现“确定”字样，单击“读取”按钮时，主站“2”上有“√”，从站“3”上是白色，没有激活的站点是灰色，这表明整个系统无故障。

**2. 用CP342－5（CP443－5）通信处理器诊断故障**

S7－300/400的常用的PROFIBUS通信处理器有CP342－5、CP342－5 FO（带光纤口）、CP343－5和CP443－5等，这些通信处理器扩展了PLC的通信接口，其通信功能比集成的DP接口的功能强大。它们支持PG/OP（编程器/控制面板）通信、S7通信和S5兼容通信（FDL）。此外，这些通信处理器还有强大的诊断功能。以下以CP342－5为例，介绍通信处理器诊断故障。




图12-100　诊断

（1）CP342－5作为主站时的诊断方法

**【例12-11】**某控制系统采用PROFIBUS通信，其硬件组态如图12-101所示，请用CP342－5诊断网络的连接状态。

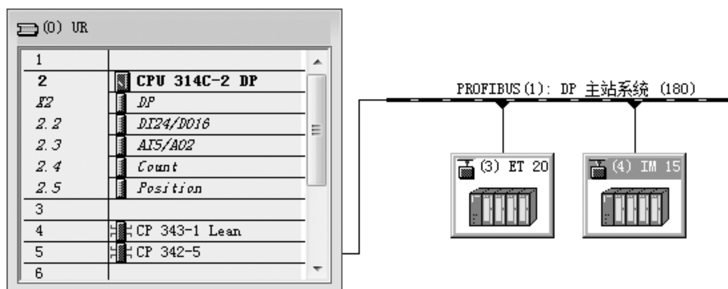



图12-101　硬件组态

**解：**

1）打开硬件组态界面，选中“CP342－5”（或者CP443－5），单击鼠标右键，单击弹出的快捷菜单中的“对象属性”子菜单，出现如图12－102所示界面，选中“诊断”选项卡，单击“运行”按钮，弹出诊断界面。




图12-102　打开诊断

2）如图12－103所示，单击“监控”按钮，单击目录树“DP主站”，可以看到界面的左侧，3号从站和4号从站上都有红色的斜杠，这是“掉站”和“站故障”的意思，接着选中“3号站的7号模块”，界面的下侧的方框中显示：“预期/实际组态不匹配”，说明3号站可能是组态错误。

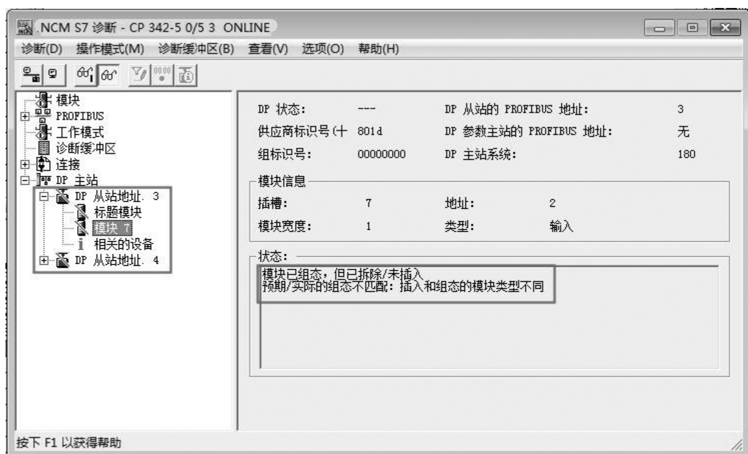



图 12-103　连接状态

3）如图 12-104 所示，单击目录树“PROFIBUS”→“节点”，可以看到界面的上方显示：2 号站（主站）处于激活状态，3 号站处于等待状态，说明 3 号站并未“掉站”。




图 12-104　节点状态

4）经检查，发现 3 号站是组态错误，重新组态下载后，再次监控，如图 12-105 所示，发现 3 号站正常，说明错误已经找到。

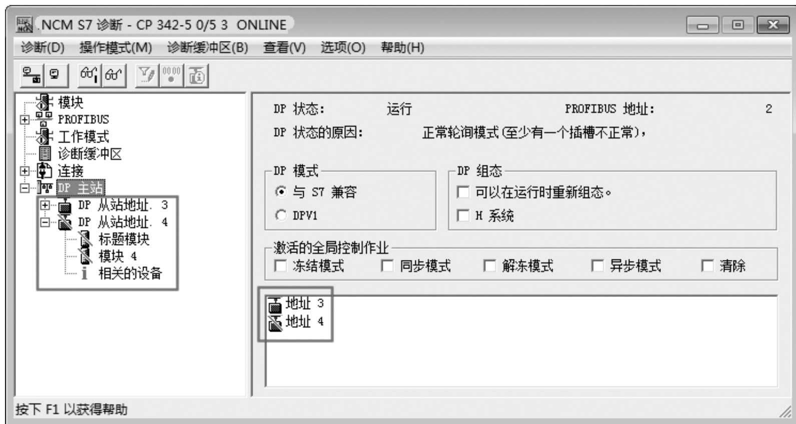



图 12-105　节点状态

5）4 号站是“掉站”，“掉站”的常见原因有可能是 DP 通信接头松动、DP 电缆断开、站点断电和从站通信模块损毁等。经检查发现是 4 号站断电。再次监控，如图 12-106 所示，发现 4 号站正常，说明故障原因已经找到了。

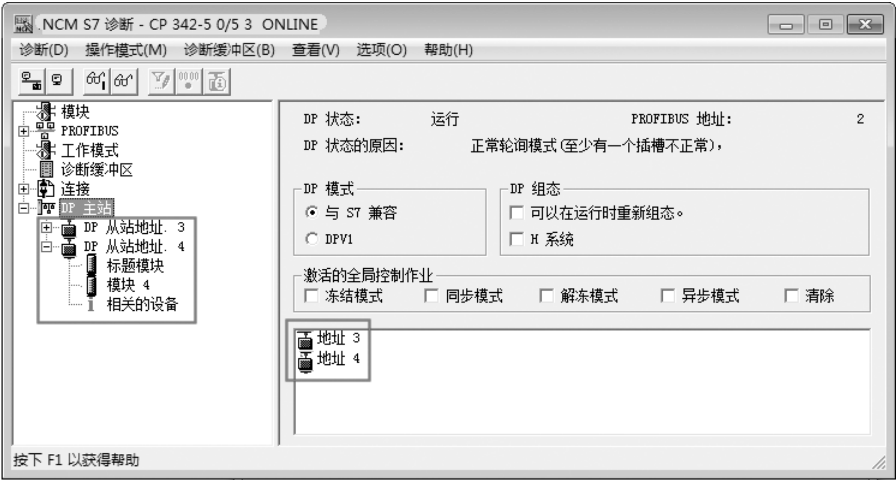



图 12-106　节点状态

（2）CP342 -5 作为从站时的诊断方法。

**【例 12-12】**某控制系统采用 PROFIBUS 通信，其硬件组态如图 12-107 所示，请用 CP342 -5 诊断网络的连接状态。

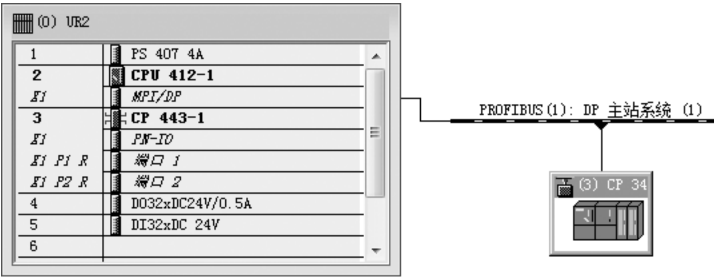



图 12-107　硬件组态

**解：**

1）打开硬件组态界面，选中“CP342 －5”（或者 CP443 －5），单击鼠标右键，在弹出的快捷菜单中选择“对象属性”子菜单，出现如图 12-108所示界面，选中“诊断”选项卡，单击“运行”按钮，弹出诊断界面。




图 12-108　打开诊断

2）如图 12-109 所示，单击“监控”按钮，单击目录树“DP 从站 3”，可以看到界面的上方 DP 状态位没有数据交换，也就是 3 号站的通信已经中断。再单击目录树“PROFIBUS3”→“节

点”，可以看到界面上方 3 号站处于激活状态，而没有看到 2 号站（主站），如图 12-110 所示，这说明可能是主站或者主站与从站的连接有问题。

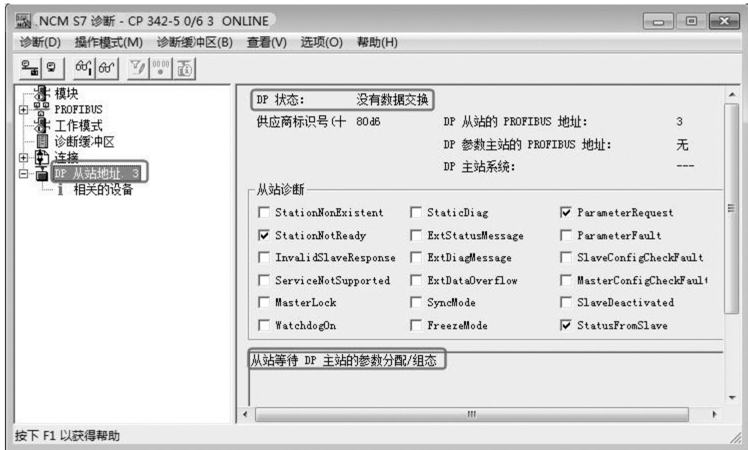



图 12-109　DP 从站连接状态




图 12-110　节点连接状态（1）

3）经检查发现是 DP 电缆断开了，修复后再监控，如图 12-111 所示，可以看到界面上方 2 号站和 3 号站都处于激活状态，说明故障原因已经找到。




图 12-111　节点连接状态（2）

### 12.2.9 用软件 Amprolyzer 诊断故障

Amprolyzer（Advanced Multicard PROFIBUS Amprolyzer，高级多卡式 PROFIBUS 总线分析仪）总线监视器是一款功能强大的软件，用于 PROFIBUS 网络诊断。在调试和维护期间，该工具对于 PROFIBUS 网络的检查具有极大帮助。Amprolyzer 也是 PROFIBUS 产品开发中的重要工具。

**1. Amprolyzer V3.2 的功能**

1）包含连接至 PROFIBUS 所有伙伴的使用寿命。

2）包含各伙伴方当前操作状态的诊断概览。

3）总线统计包含事件数，例如超时和消息帧重复。

4）自动识别传输速率。

5）带有触发器和过滤器属性（基于事件和消息内容，包括时间标记）的消息记录。

6）保存并以 Excel 格式输出消息记录。

**2. Amprolyzer 软件的安装条件**

Amprolyzer V3.2 可以在西门子官网上免费下载，其仅支持 Windows2000/Windows XP Professional 操作系统。

**3. Amprolyzer 软件的故障诊断方法**

Amprolyzer V3.2 结合高级的 PROFIBUS 板卡，是一款用于 PROFIBUS 网络诊断的强大工具软件，以下介绍其使用方法。

1）起动 Amprolyzer，单击“Add Bus…”按钮，如图 12-112 所示，将弹出“Add Bus…”对话框。

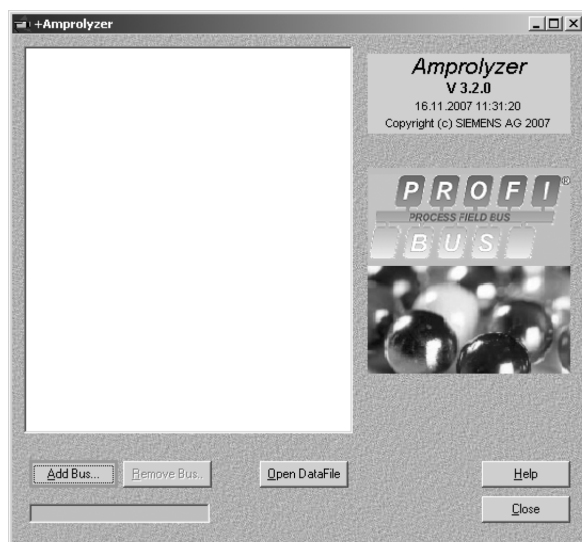



图 12-112　Amprolyzer 初始画面

2）选择 PROFIBUS 接口。例如 CP5611，如图 12-113 所示，单击“OK”按钮。Amprolyzer 仅能够支持 CP5511、CP5512 和 CP5611 卡，且必须电脑中已经安装这些硬件中的一种。

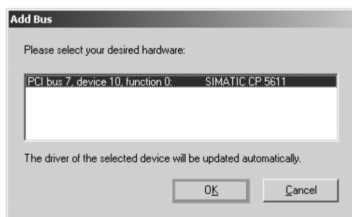



图 12-113　选择 PROFIBUS 接口

3）创建触发事件。在 Amprolyzer 中，双击“Record（Complex）”选项，如图 12-114 所示，将弹出“Record（Complex）”对话框 。

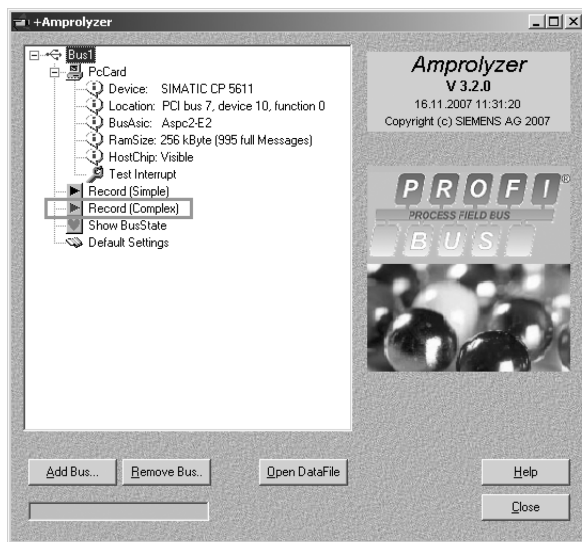



图 12-114　创建触发事件

4）设置触发参数。在“Record（Complex）”对话框中选择“Events”选项，可以设置事件触发的一些参数。

在“Msg Events”复选框中可以选择哪些参数将触发消息事件，如重复请求、祯错误。

在“Action”选项框中可以选择消息事件的动作，如图 12-115 所示。

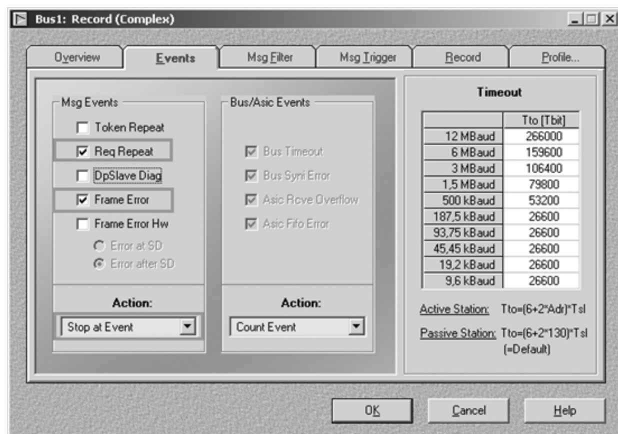



图 12-115　设置触发事件参数

5）设置触发消息。在“Record（Complex）”对话框中，选择“Msg_Trigger”选项框，选择事件触发的动作（例如：停止记录、起动记录和在 x 个消息后停止记录），在本例中，当产生了 2000 条消息后停止记录，如图 12-116 所示。

另外，还需要定义事件触发的条件，共有两个选项：

第一个选项选择“DSAP. dec”，即目标访问点，以十进制表示。

第二个选项有如下子项：

- 60(dec) = 诊断消息。
- 61(dec) = 参数消息(节点失效)。
- 62(dec) = 组态消息。

本例选择“061”。此外，还有其他 3 种的触发条件，在此不再赘述。

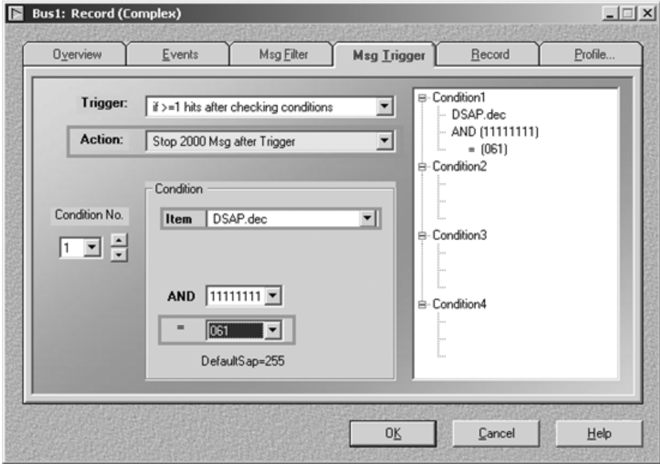



图 12-116　设置触发消息

6）定义被记录消息的最大长度。在“Record（Complex）”对话框中，选择“Record”选项框，定义被记录消息的最大长度，本例中设置为 50 B。之后单击“OK”按钮，开始起动诊断事件记录，如图 12-117 所示。

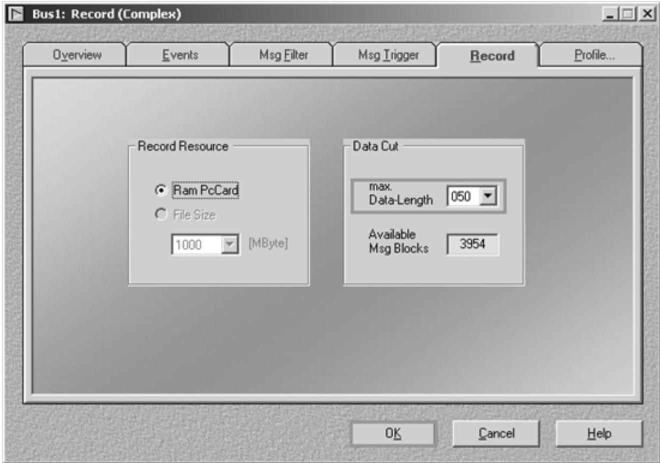



图 12-117　定义被记录消息的最大长度

注意：CP 卡对消息有限制，消息长度的限制可以通过增加消息号来弥补。诊断数据可能只是消息截留后的一部分。

7）诊断结果解读。如果显示“Bus active”状态，如图 12-118 所示，则意味着已起动记录，将记录事先定义的事件。右侧的方格中显示已经激活的站点，没有激活的站点不在其中显示，本例只显示了“2”号站。

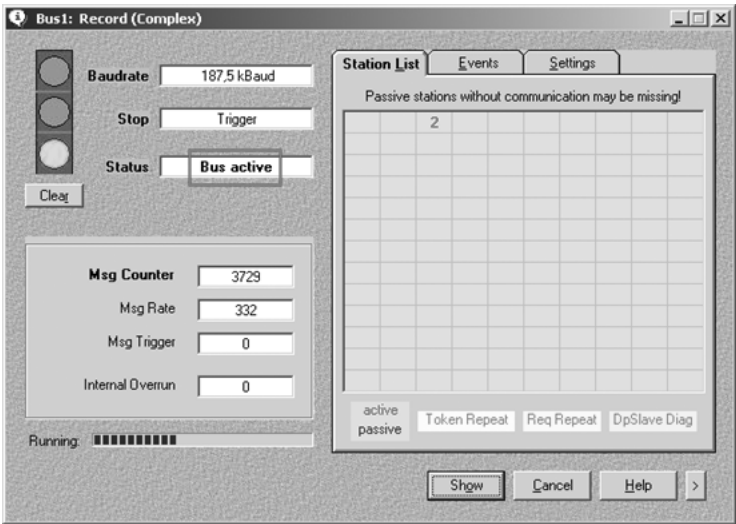



图 12-118　诊断结果

## 12.3　各种故障诊断的比较

各种故障诊断的比较情况如下：

（1）使用设备上的 LED 进行诊断

简单、方便、直观，但给出的故障很笼统，需要其他方法（如 STEP 7 的诊断视图、诊断视图和模块信息等）配合诊断，才能得出比较准确的诊断信息。如果系统比较大，查看诊断信息，也比较费事。

（2）使用 STEP 7 进行诊断

简便易行，可以迅速获得准确、详细的诊断信息，CPU 的模块的诊断缓冲区提供了错误的文本信息，例如 DP 站地址、出错模块的地址故障等。但需要安装 STEP 7 软件，并会使用 STEP 7 软件进行故障诊断的操作方法。

（3）使用 OB86 和 OB82 的局部变量进行诊断

产生故障后，调用 OB86、OB82，调用 SFC20 保存 OB 的 20B 局部变量后，可获取产生的故障信息，查阅 OB 的在线帮助，获得局部变量的意义，即是诊断信息。

（4）使用硬件进行诊断

1）PLC 的通信处理器进行诊断，诊断信息不十分具体。

2）诊断中继器进行诊断，可以诊断故障的具体位置，但组态和诊断的准备工作比较复杂。

3）BT－200 总线诊断器诊断。多用于系统的安装和调试过程。

（5）调用 SFC13 进行诊断

诊断 DP 网络故障最常用的是 SFC13。在分析和编写程序时，必须查询相关手册，要搞清楚每个字节的含义，数据分析工作量大，要求相对较高，有一定的难度。

（6）调用 FB126 进行诊断

FB126 是中断诊断驱动的功能块，其背景数据块有 1000 多个字节，可以用变量表示各种状态的从站。分析 FB126 提供的诊断数据量的工作比较大。

（7）用报告系统错误功能诊断和显示故障

STEP 7 的“报告系统错误”功能只需要简单组态，几乎全部采用默认参数就可以自动生成用于诊断和发送消息的 OB、FB、SFC 和 DB，以及各种机架、从站和模块的故障信息，故障的消息信息传送到 HMI 或者 WinCC，用消息的形式显示故障。

（8）其他的故障诊断方法

还有诊断中继器、SFC51 等诊断方法。

总之，故障诊断比较复杂，不易掌握，读者应该由浅入深，先掌握 S7－300/400 的一些基础知识，再学习用本章介绍的方法，诊断 PLC 故障。

## 12.4　故障诊断的显示

通过以上故障诊断的讲解可知，故障诊断的方法虽然很多，但完全掌握比较困难，而且需要较多的专业知识为基础。因此，一些刚入职的技术人员特别希望的是系统发生故障后，与故障相关的信息能直接显示到安装了 WinCC 软件的计算机上或者触摸屏上，这样只要通过查看故障信息就可以快速找到故障点。

要实现所有的详细故障都能显示到人机界面（HMI）上是不现实的，因为一方面需要程序设计者编写大量的程序，工作量极大；另一方面，现场具体情况复杂多变，再优秀的程序设计者也不可能把所有可能发生的故障都编写在程序中。但是毫无疑问，一个优秀的程序设计者，会尽可能把现场可能的故障情况考虑进去。

消息是用来向操作人员报告系统当前的运行状态、故障位置及原因，以提供快速检测、定位和排除故障，减少设备的停机时间。组态时，用 STEP 7 创建、编辑和编译消息，运行时，PLC 将消息自动发送到 WinCC 和 HMI，并显示故障消息。

消息的发送分为位发送和消息号发送两种方法。以下用几个例子来介绍故障显示的编程方法。

### 12.4.1　位发送故障显示

当 PLC 和 HMI 没有公用的数据库时采用位消息发送，如 PLC 是西门子的产品，而 HMI 是 MCGS 的产品，这时就没有公用数据库。如果用 HMI 组态软件 WinCC flexible 对位消息传送组态，则用 PLC 中的位信号作为触发信号。运行时，HMI 不断查询位状态，位为 1 时，表示有故障，HMI 的报警视图或者报警窗口显示报警文本。位为 0 时，表示故障消失，HMI 的报警视图或者报警窗口显示报警消失。因为 HMI 需要不断扫描 PLC 中故障位状态，所以

通信负荷重。

**【例 12-13】**某控制系统，其主要控制模块由 CPU314C-2DP、IM53-1 和 SM331 等组成，采用 PROFIBUS-DP 通信，要求发生通信故障时，HMI 接收通信故障报警，请实现此功能。

**解：**

1）用 STEP 7 新建项目。命名为“ALARM1”，在硬件组态界面中，先后插入机架“RACK”和 CPU314C-2DP。再新建 PROFIBUS-DP 网络，插入从站模块 IM153-1 和模拟量模块 SM331。硬件组态如图 12-119 所示。

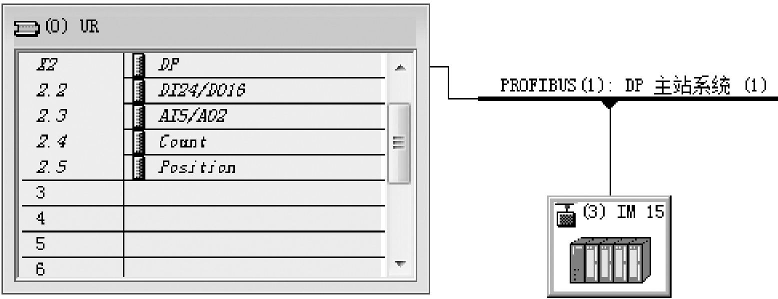



图 12-119 硬件组态

2）编写报警程序。在 SIMATIC Manager 中，选中“块”，在块中插入通信故障组织块 OB87，打开 OB87，编写程序如图 12-120 所示。

```
       SET
       SAVE
       =     L     20.1
       A     M      0.5
       NOT
       JCN   M001
       SET
       =     M      0.5
M001:  CLR
       A     L     20.1
       SAVE
       BE
```

图 12-120 OB87 中的程序

以上指令表程序的功能是：当发生通信故障时，执行 OB87 中的程序，使 M0.5 得电。因为 M0.5 和 HMI 关联，当发生通信故障时，可以发出通信报警信息，显示在 HMI 上。

3）用 WinCC flexible 新建项目。命名为“ALARM1”。

4）新建连接。如图 12-121 所示，通信程序为“SIMATIC S7 300/400”，HMI 的地址为“1”，PLC 的地址为“2”，HMI 和 PLC 的通信配置文是“MPI”。

5）新建变量。如图 12-122 所示，新建变量为“MW0”，数据类型为“WORD”。

6）设置触发报警变量。如图 12-123 所示，“MW0”的第 13 位，即是“M0.5”是触发变量。特别注意，不是第 5 位。

7）新建画面。如图 12-124 所示，在工具的“增强对象”选项中，将“报警视图”控




图 12-121　新建连接




图 12-122　新建变量




图 12-123　设置报警触发变量

件拖入画面。

8）运行仿真。当 PLC 发生通信故障时，M0.5 置位，由于 HMI 循环扫描 PLC，当 PLC 扫描到 M0.5 置位，触发报警，如图 12-125 所示。

## 12.4.2　消息号发送故障显示

当 PLC 和 HMI 有公用的数据库时，采用消息号发送故障信息。组态和编程时，将消息分配给用户程序中的某个位。PLC 的状态改变时，只将对应的消息号传送到 HMI，显示出保存在 HMI 或者 WinCC 中的消息文本，消息使用 PLC 的时间标记，这种方法的通信减轻了通信负荷。

**【例 12-14】** 某控制系统，其主要控制模块由 CPU314C-2DP 和 SM331 等组成，要求发生故障时，PLC 触发 HMI 故障报警，请实现此功能。

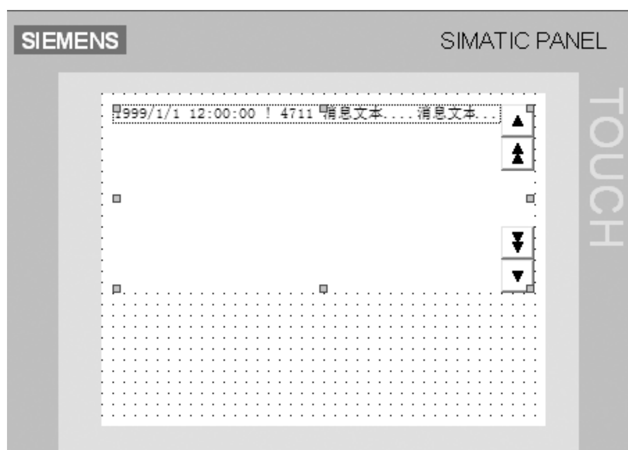



图 12-124　新建画面

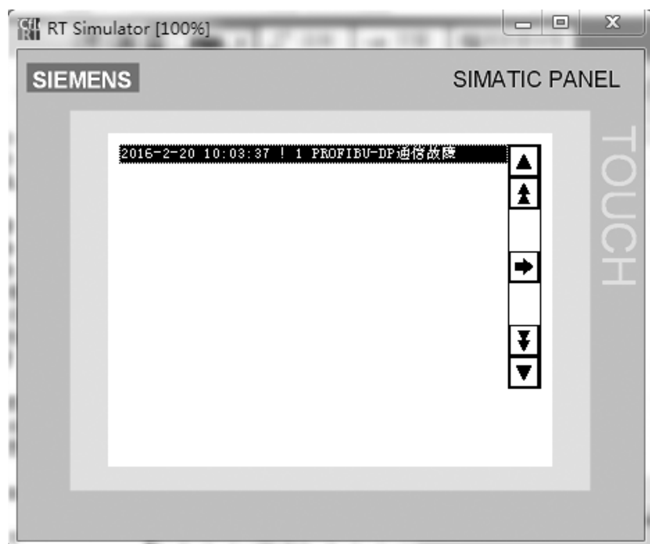



图 12-125　报警画面

**解：**

1）用 STEP 7 新建项目。命名为“ALARM2”，先在 SIMATIC Manager 界面的菜单中，单击“插入”→“站”→“SIMATIC S7300”，然后打开硬件组态界面，并在其中先后插入机架“RACK”和 CPU314C-2DP，再插入模拟量模块 SM331。保存和编译后回到 SIMATIC Manager 界面，再单击“插入”→“站”→“SIMATIC HMI-Station”，选择 HMI 的型号为“TP 177B 6″ color PN/DP”，如图 12-126 所示。最后 SIMATIC Manager 界面如图 12-127 所示。

2）在 NetPro 中建立连接。实现 HMI 和 PLC 连接的方法如下：在 SIMATIC Manager 界面，单击“网络组态”按钮，打开网络组态工具“NetPro”，界面中显示 HMI 和 PLC 之间 MPI 并未连接。用鼠标的左键将两个站代表 MPI 接口的红色小方框拖拽到 MPI 网络上，即可实现两个站的连接，如图 12-128 所示。单击工具栏上的“保存并编译”按钮。

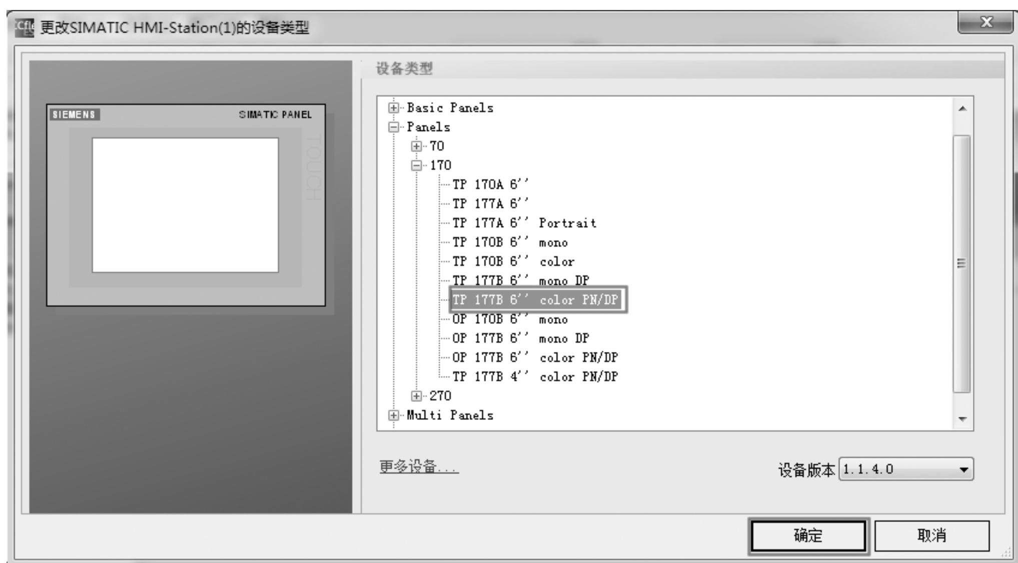



图 12-126　选择 HMI 的型号




图 12-127　SIMATIC Manager 界面

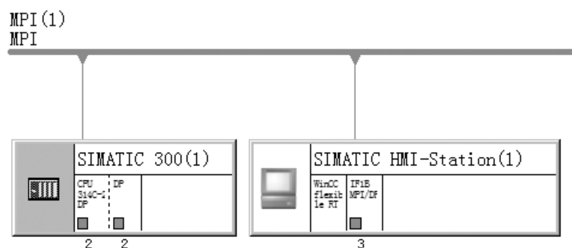



图 12-128　NetPro 中建立连接

3）编写功能块 FB1 的程序。在 SIMATIC Manager 界面的菜单中，创建一个功能块 FB1，双击之，打开 FB1。

在 FB1 的局部变量表中创建一个双字型（Dword）的输入（IN）参数 EV_ID1，如图 12-129 所示。单击鼠标右键，弹出快捷菜单，单击“对象属性”，弹出变量属性对话框，如图 12-130 所示，选中“属性”选项卡，并作如图 12-130 所示的设置，单击“确定”按钮。用同样的方法生成输入变量 EV_ID2。

在 FB1 的局部变量表中创建三个双字型（Dword）静态变量 SD1、SD2 及 SD3，用于保存消息的相关值，它可作为消息的组成部分发送到 HMI。

内容：'环境\接口\IN'

接口：IN（EV_ID1、EV_ID2）、OUT、IN_OUT、STAT（SD1、SD2、SD3）、TEMP

| 名称 | 数据类型 | 地址 | 初始值 | 排除地址 | 终端地址 | 注释 |
|---|---|---|---|---|---|---|
| EV_ID1 | DWord | 0.0 | DW#16#0 | □ | □ | |
| EV_ID2 | DWord | 4.0 | DW#16#0 | □ | □ | |
| | | | | □ | □ | |

图 12-129　声明局部变量

变量属性

常规 | 信息 | 属性

| | 属性 | 数值 |
|---|---|---|
| 1 | S7_a_type | alarm_s |
| 2 | S7_server | alarm_archiv |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | | |

注意：

插入行(I)　删除行(D)　确定　取消　帮助

图 12-130　变量属性

在 FB1 中，编写如图 12-131 所示的程序。

```
程序段 1：标题：
L     8.020000e+001
T     #SD1                    #SD1
L     3.925000e+002
T     #SD2                    #SD2
L     MW    50
T     #SD3                    #SD3
CALL  "ALARM_SQ"              SFC17
 SIG    :=M10.0
 ID     :=W#16#EEEE
 EV_ID  :=#EV_ID1             #EV_ID1
 SD     :=P#DB1.DBX8.0 BYTE 12
 RET_VAL:=MW12

程序段 2：标题：
L     2354
T     #SD1                    #SD1
L     7.536000e+002
T     #SD2                    #SD2
L     MW    52
T     #SD3                    #SD3
CALL  "ALARM_S"               SFC18
 SIG    :=M10.1
 ID     :=W#16#EEEE
 EV_ID  :=#EV_ID2             #EV_ID2
 SD     :=P#DB1.DBX8.0 BYTE 12
 RET_VAL:=MW16
```

图 12-131　FB1 中的程序

4）创建消息文本。在 SIMATIC Manager 界面中，选中功能块 FB1，单击右键，弹出快捷菜单，选择“特殊对象属性”→“消息”，打开“消息组态”对话框，如图 12-132 所示。

单击“更多》”按钮，选中“EV_ID1”，在“消息文本”方框中输入“1 号电机 电流：@1R%5.1f@ 电压：@2R%5.1f@”，在信息文本中输入“电机过载，电流过大”；再选中“EV_ID2”，在“消息文本”方框中输入“2 号电机 电流：@1X%5u@ 电压：@2R%5.2f@”，在信息文本中输入“电机过载，电流过大”，最后单击“确定”按钮，如图 12-133 所示。




图 12-132　消息组态（1）




图 12-133　消息组态（2）

SFC 用参数 SD 指定的地址来传递相关值。消息文本要遵循一定的格式，以下简要介绍。

① 消息文本的格式：

@ <相关值的编号><元素类型><格式代码>@

以上格式中，“@”是字符串的起始和结束标识。

② 元素类型。B、Y、C、W、I、X、D 和 R 分别用来表示位（BOOL）、字节（BYTE）、字符串、字（WORD）、整数（INT）、双字（DWORD）、双整数（DINT）和浮点数（REAL）。

③ 格式代码。格式码见表 12-20。

**表 12-20　格式码**

| 序　　号 | 格 式 代 码 | 含　　义 |
|---|---|---|
| 1 | % iX | i 位十六进制数 |
| 2 | % iu | i 位无符号十进制数 |
| 3 | % id | i 位有符号十进制数 |
| 4 | % ib | i 位二进制数 |
| 5 | % is | i 位 ANSI 字符串 |
| 6 | % i. yf | 小数点后 y 位，共 i 位的定点数 |
| 7 | % t# | 访问文本库 |

举例，“@2R%5. 1f@”表示序号 2，浮点数，显示最多 5 位，显示 1 位小数；“@1X%5u@”表示序号为 1，5 位双字十进制显示。

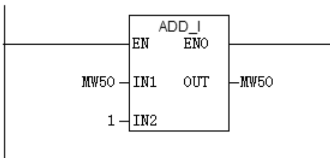
此外，以上所有的标点符号都应该在“英文”状态下输入，而不能在“汉字”状态下输入。

5）在 OB1 中调用 FB1。

```
CALL  FB       1,DB1
  EV_ID1 = DW#16#60000001
  EV_ID2 = DW#16#60000002
```

6）编写 OB35 中的程序，如图 12-134 所示。

⊟ **程序段 1**：标题：




⊟ **程序段 2**：标题：

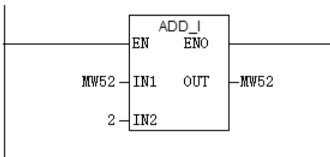



图 12-134　OB35 中的梯形图

7）HMI 显示消息相关设置。

① 创建画面。在 SIMATIC Manager 界面中，单击“SIMATIC HMI - Station（1）”→“WinCC flexible RT”→“画面”，如图 12-135 所示，再把右侧“工具栏”中的“报警视图”控件用鼠标拖拽到 HMI 的“画面_1”中。

② 激活连接。如图 12-136 所示，双击画面右侧的“通信”中的“连接”图标，弹出“连接”对话框，单击“激活的”下方的倒三角，选择“开”选项，这样原来的“关”选项变为“开”，HMI 与 PLC 就建立起连接了。

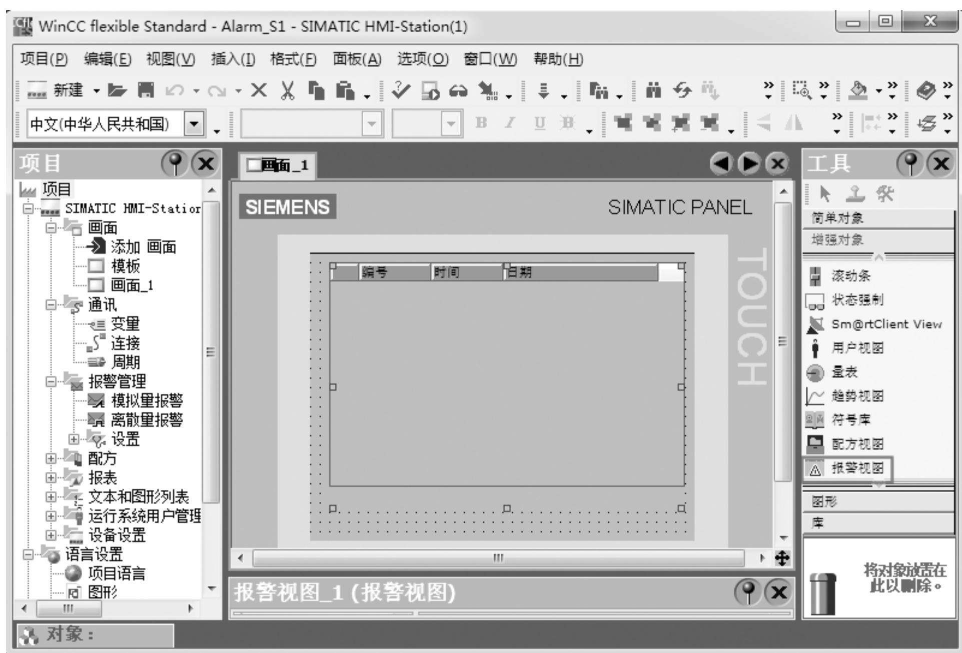



图 12-135　创建画面




| 名称 | 激活的 | 通讯驱动程序 | 工作站 | 伙伴站 | 节点 |
|---|---|---|---|---|---|
| 连接_1 | 开 | SIMATIC S7 300/400 | \Alarm_S1\SI… | CPU 314C-2 DP | CPU 314C-2 DP |

图 12-136　激活连接

③ 设置报警。如图 12-137 所示，展开目录树“报警管理”→“设置”→“报警设置”，选中“报警设置”并双击之。在“系统报警”中，勾选“S7 诊断报警（编号）”和“S7 诊断报警（文本）”；在“报警程序”表格的第一行，勾选“所有显示类”，最后单击右下方的“√”按钮。

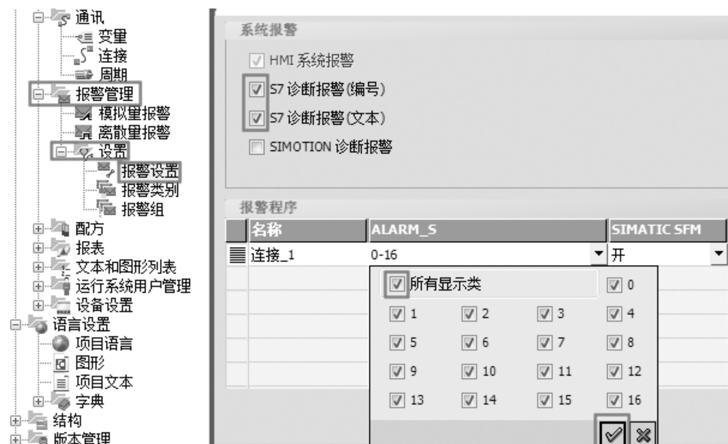



图 12-137　设置报警

④ 组态报警视图。双击目录树中的“画面_1”，选中“画面_1”中的“报警视图”控件，再选择“常规”选项→“报警事件”→“S7 报警”，如图 12-138 所示。保存所有的设置。




图 12-138　组态报警视图

⑤ 运行仿真。单击工具栏上的“起动运行系统”按钮，HMI 的仿真运行开始，因为 PLC 没有向 HMI 发送报警触发信号，此时 HMI 不会报警。

在 SIMATIC Manager 界面中，单击“打开/关闭仿真器”按钮，并将 PLC 的程序下载到仿真器中，这部分内容前面章节已经讲述，在此不再赘述。将 PLC 的仿真器置于“RUN”状态，并将 M10.0 和 M10.1 置位，如图 12-139 所示，M10.0 和 M10.1 置位表示激活了 1 号报警和 2 号报警，这点可以在程序 FB1 中得知。

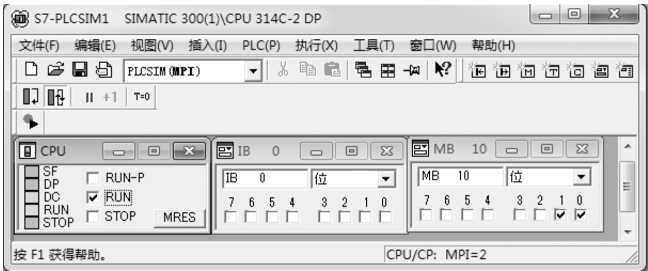



图 12-139　PLC 运行仿真

由于，M10.0 和 M10.1 置位表示激活了 1 号报警和 2 号报警，可以看到如图 12-140 所示的画面。

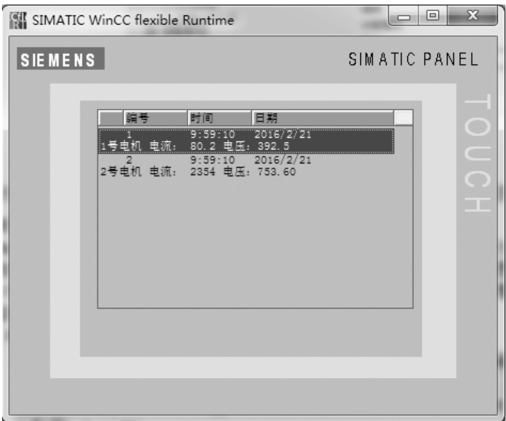



图 12-140　HMI 运行仿真

## 12.4.3　用 S7－PDIAG 软件组态过程诊断

**1. S7－PDIAG 软件简介**

S7－PDIAG 软件在内部调用系统功能 SFC17/SFC18、SFC107/SFC108 来完成消息传送，与前面的直接调用 SFC 相比，S7－PDIAG 软件具有通信资源管理功能，因此功能更加强大，而且使用更加简单。使用 S7－PDIAG 软件，实际是利用消息号发送故障信息。

（1）S7－PDIAG 软件的安装

S7－PDIAG 软件不能离开 STEP 7 独立运行，必须安装了 STEP 7 后才能安装此插件。S7－PDIAG 软件不是 STEP 7 的自带软件，可在西门子的官网上下载安装，但需要购买授权。

不同的 S7－PDIAG 版本与操作系统有兼容性问题，简单来说如果 S7－PDIAG 的版本较低（早期的版本）可能无法安装到目前的操作系统上，例如 S7－PDIAG V5.3 SP1 不能安装在 Win2008 R2（64）操作系统中。S7－PDIAG 与操作系统的兼容性见表 12–21。

**表 12–21　S7－PDIAG 与操作系统的兼容性**

| 产品版本 | STEP 7 V5.5 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | WinXP sp2 | WinXP sp3 | Win7 | Win7 sp1 | Win7（64） | Win7 sp1（64） | Win2008 R2（64） | Win2008 R2 SP1（64） |
| S7－PDIAG V5.1 | X | – | – | – | – | – | – | – |
| S7－PDIAG V5.3 SP3 | X | X | – | – | – | – | – | – |
| S7－PDIAG V5.3 SP5 | X | X | X | – | – | – | – | – |
| S7－PDIAG V5.3 SP6 | X | X | X | X | X | X | X | X |

注：1. 表 12–20 中的“X”表示兼容。
2. “－”表示不兼容。
3. 举例：Win7(64)操作系统需要安装 S7－PDIAG V5.3 SP6 或以上版本的软件包。

（2）S7－PDIAG 软件的功能

S7－PDIAG 软件包含三种诊断功能，分别是地址监控（Address monitoring）、全局监控（Global monitoring 或 General monitoring）和运动监控（Motion monitoring））。

1）地址监控。地址监控功能就是通过监控一个位号的状态变化，可以设定一个沿电平信号（本例为下降沿电平信号），如图 12–141所示。

2）全局监控。全局监控也叫常规监控。全局监控功能监控一个表达式，如果表达式的结果为 1，则触发消息。全局监控如图 12–142 所示。

在“初始诊断地址”中，设置初始诊断地址，如 M20.0。在“监控定义”中，定义时间触发条件，触发条件必须用 S7－PDIAG 语言编写，S7－PDIAG 语言包含的指令格式见表 12–22。

**表 12–22　S7－PDIAG 语言包含的指令格式**

| 序　号 | 指　令 | 含　义 |
|---|---|---|
| 1 | AND | 与操作，例：M1.1 AND M1.3 |
| 2 | ONDT | ON 信号的延时操作。例：ONDT（M0.0，T#1S），表示当 M0.0 为 1 时，延时 1 s 后表达式为 1 |

（续）

| 序　　号 | 指　　令 | 含　　义 |
|---|---|---|
| 3 | NOT | 取反操作。例：NOT M0.0，表示 M0.0 为 1 时，表达式为 0 |
| 4 | EN | 存入下降沿结果。例：EN（M0.0），表示 M0.0 下降沿时表达式为 1 |
| 5 | EP | 存入上升沿结果。例：EP（M0.0），表示 M0.0 上升沿时表达式为 1 |
| 6 | OR | 或操作。例：M1.1 OR M1.3 |
| 7 | SRT | 置位复位延时操作。例：SRT（M1.1，M1.2 T#2S），表示如果 M1.1 为 1，2 s 后表达式为 1，如果 M1.1 再次为 1，表达式为 0，2 s 表达式为 1，如果 M1.2 为 1，表达式为 0 |
| 8 | XOR | 异或操作。例：M1.1 XOR M1.3 |

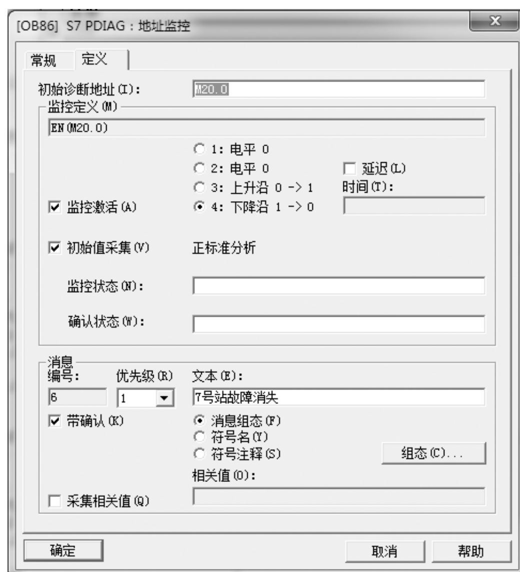



图 12-141　地址监控




图 12-142　全局监控

3）运动监控。运动监控是对用户编写的定位过程进行监控，运动监控有 4 种监控模式，分别是执行监控模式（Action Monitoring）、起动监控模式（Startup Monitoring）、反应监控模式（Reaction Monitoring）和互锁监控模式（Interlock Monitoring）。

**2. S7－PDIAG 软件应用举例**

**【例 12-15】**某控制系统，其主要控制模块由 CPU314C－2DP 和 SM331 等组成，要求发生故障时，PLC 触发 HMI 故障报警，请用 S7－PDIAG 软件实现此功能。

**解：**

1）用 STEP 7 新建项目。命名为“ALARM3”，在 SIMATIC Manager 界面的菜单中，单击“插入”→“站”→“SIMATIC S7300”，打开硬件组态界面，并在其中先后插入机架“RACK”和 CPU314C－2DP，再插入模拟量模块 SM331，新建 PROFIBUS 网络，然后插入从站“IM153－1”，地址为 3，从站上插入 SM323 模块，硬件组态如图 12-143 所示。保存和编译后回到 SIMATIC Manager 界面，再单击“插入”→“站”→“SIMATIC HMI－Station”，选择 HMI 的型号为 TP 177B6″ color PN/DP，如图 12-144 所示。最后界面如图 12-145 所示。

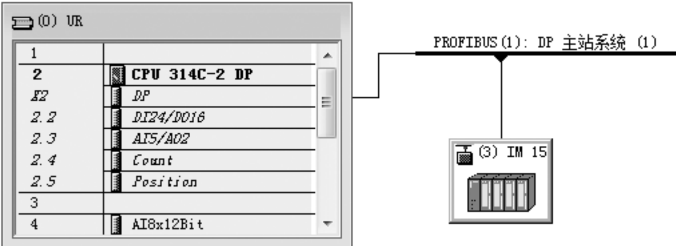



图 12-143　选择 HMI 的型号

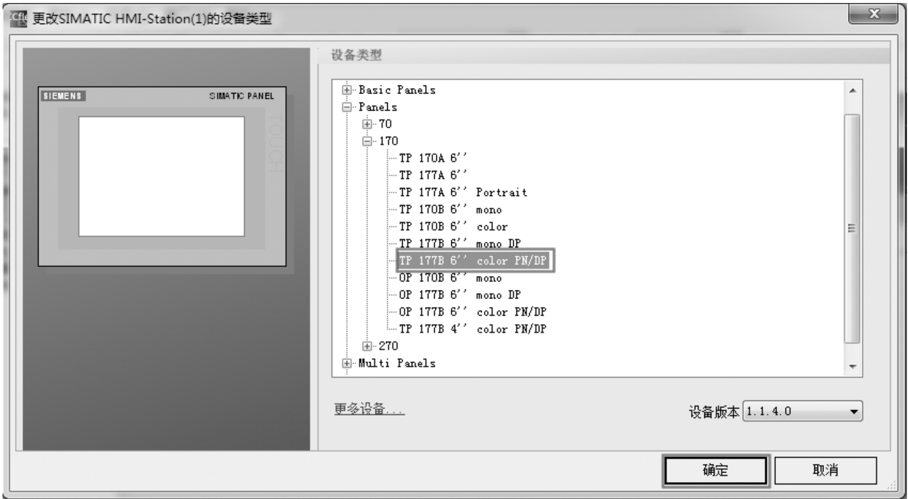



图 12-144　选择 HMI 的型号

2）在 NetPro 中建立连接。实现 HMI 和 PLC 连接的方法如下：在 SIMATIC Manager 界面，单击“网络组态”按钮，打开网络组态工具 NetPro，界面中显示 HMI 和 PLC 之间 MPI 并未连接。用鼠标的左键将两个站代表 MPI 接口的红色小方框拖拽到 MPI 网络上，即可实




图 12-145 SIMATIC Manager 界面

现两个站的连接，如图 12-146 所示。单击工具栏上的“保存和编译”按钮。

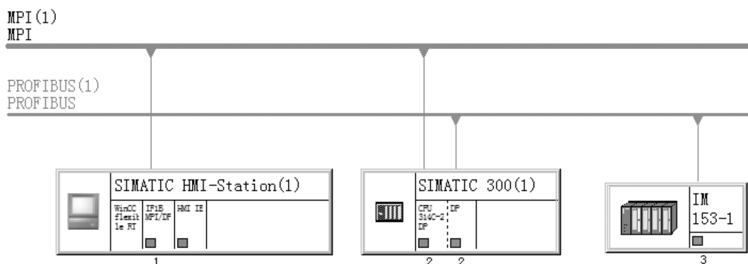



图 12-146 NetPro 中建立连接

3）编写 OB86 中的程序。在 SIMATIC Manager 界面中，选中“块”，并在“块”中插入组织块 OB86，双击打开 OB86，编写程序如图 12-147 所示，程序的含义是：当出现机架故障时，M10.0 为 1；当出现机架故障消失时，M10.0 为 0。

```
      SET
      SAVE
      =     L     20.1
      L     #OB86_EV_CLASS
      L     B#16#39
      ==I
      JCN   M001
      SET
      =     M     10.0
      JU    M002
M001: L     #OB86_EV_CLASS
      L     B#16#38
      ==I
      JCN   M002
      CLR
      =     M     10.0
M002: CLR
      A     L     20.1
      SAVE
      BE
```

图 12-147 OB86 中的程序

4）组态地址监控。在程序编辑中，打开 OB86，选中 M10.0，单击鼠标右键，弹出快捷菜单，选择“特殊对象”→“监视”，弹出如图 12-148 所示界面，选中“S7 PDIAG：地址监控”；再单击“新建”按钮，弹出如图 12-149 所示的界面，在“定义”选项卡中，进行如图 12-149 所示的设置，设置完成后，单击“确定”按钮，含义是：当 M10.0 上升沿时，

触发 HMI 报警，报警文本是“3 号机架故障”；用同样的方法定义当 M10.0 下降沿时，触发 HMI 报警，报警文本是“3 号机架故障消失”，如图 12-150 所示。




图 12-148　过程监控

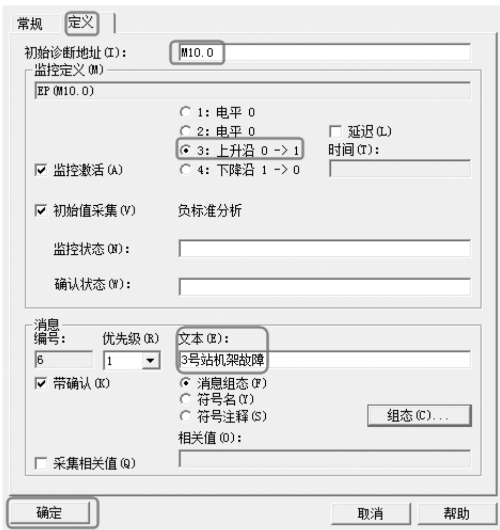



图 12-149　地址监控（1）

5）组态过程诊断。在 SIMATIC Manager 界面的菜单中，单击菜单栏的“选项”→“组态过程诊断”，弹出如图 12-151 所示界面；单击“编译”按钮，如果没有错误，则图中 S7 程序（1）处的红色叉号消失。编译后，SIMATIC Manager 的“块”中自动生成一些程序块，这些程序块与故障诊断相关。

6）编写 OB1 中的程序。编写程序如图 12-152 所示。

7）HMI 显示消息相关设置。

① 创建画面。在 SIMATIC Manager 界面中，单击“SIMATIC HMI - Station(1)”→“WinCC flexible RT”→“画面”，如图 12-153 所示，再把右侧“工具栏”中的“报警视图”控件用鼠标拖拽到 HMI 的“画面_1”中。

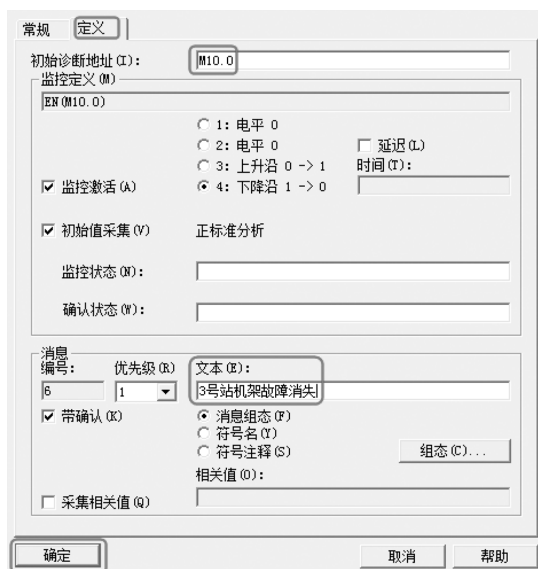



图 12-150　地址监控（2）

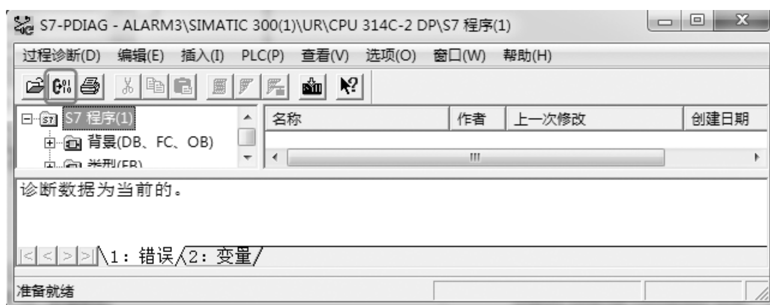



图 12-151　组态过程诊断

⊟ **程序段 1**：标题：

```
CALL  FB    44 , DB44
 PDIAGZyklus:=#OB1_SCAN_1
```

图 12-152　OB1 中的程序

② 激活连接。如图 12-154 所示，双击画面右侧“通信”中的“连接”图标，弹出“连接”对话框，单击“激活的”下方的倒三角，选择“开”选项，这样原来的“关”选项变为“开”，HMI 与 PLC 就建立起连接了。

③ 设置报警。如图 12-155 所示，展开目录树“报警管理”→“设置”→“报警设置”，选中“报警设置”，并双击之。然后，在“系统报警”中，勾选“S7 诊断报警（编号）”和“S7 诊断报警（文本）”；在“报警程序”表格的第一行，勾选“所有显示类”，最后单击右下方的“√”按钮。

④ 组态报警视图。双击目录树中的“画面_1”，选中“画面_1”中的“报警视图”控件，再选择“常规”选项→“报警事件”→“S7 报警”，如图 12-156 所示。保存所有的

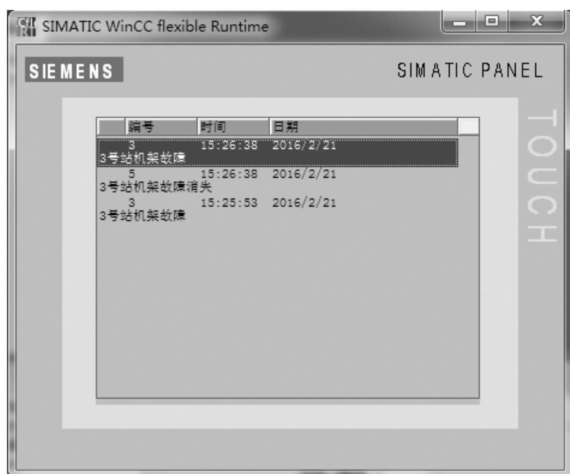



图 12-153　创建画面




图 12-154　激活连接

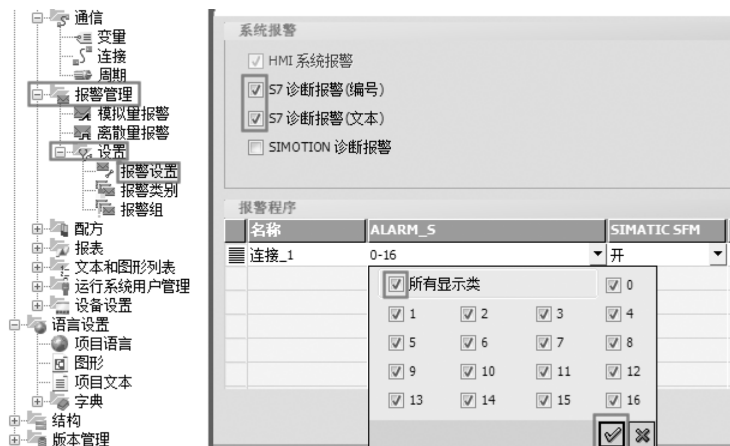



图 12-155　设置报警

设置。

⑤ 运行仿真。单击工具栏上的“起动运行系统”按钮，HMI 的仿真运行开始，因为 PLC 没有向 HMI 发送报警触发信号，此时 HMI 不会报警。

在 SIMATIC Manager 界面中，单击“打开/关闭仿真器”按钮，并将 PLC 的程序下载到仿真器中，这部分内容前面章节已经讲述，在此不再赘述。将 PLC 的仿真器置于“运行”状态，并将 M10.0 置位，如图 12-157 所示，M10.0 置位表示激活了机架故障报警，M10.0 复位表示激活了机架故障报警消失。

由 M10.0 置位和复位激活的报警信息显示在 HMI，如图 12-158 所示。




图 12-156　组态报警视图

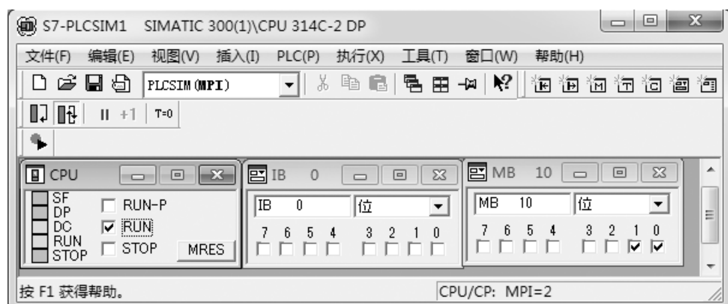



图 12-157　PLC 运行仿真




图 12-158　HMI 运行仿真

## 12.5　工程常见调试和故障诊断 30 例

以下有 30 个故障诊断实例，故障涉及 PLC、变频器、伺服驱动和通信，是作者在调试和维修设备时碰到的典型问题，具有代表性。

**【调试和故障诊断实例 1】**

故障现象：

某冲压设备，控制系统由 PLC、伺服和变频器组成，运料小车由伺服驱动同步带驱动。

使用一段时间后，发现同步带松弛，张紧同步带后发现，定位不准。

故障检查与排除：

1）这是正常现象，同步带松弛，张紧后，同步带的节距变长，导致定位不准。

2）解决方案：需要重新定位。使用伺服系统和同步带定位时，必须要设计重新定位的程序和操作界面。

**【调试和故障诊断实例 2】**

故障现象：

某钻杆热处理设备，控制系统由 PLC 和变频器组成，变频器和电动机驱动丝杠旋转，从而使运料小车作往复运动。工艺要求送料小车前进时慢速（20 Hz），后退时快速（90 Hz），且要求前进到位后尽快返回。使用一段时间后，发现返程时，小车经常超程，导致损害设备。

故障检查与排除：

1）最容易想到的办法是：加长停机时的减速时间，但本例的工艺不允许修改启停时间。

2）解决方案：在后退的限位开关前，加一个限位开关，变频器接到此信号后，减速到（20 Hz），然后停下，问题解决。很显然，以前的系统设计有缺陷。

**【调试和故障诊断实例 3】**

故障现象：

图 12-159 是某初学者设计的控制系统，使用几天后，变频器电源烧毁。

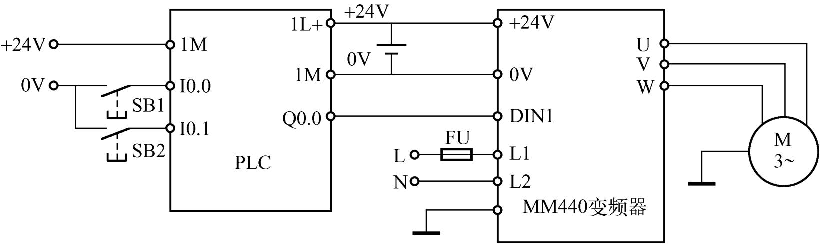



图 12-159　原理图

故障检查与排除：

1）如图 12-159 所示的原理图有错误。错误在于变频器的 DIN1 上有两个供电电源，一个是外部电源，一个是变频器内部电源，通常这是不允许的。

2）解决方案：方案 1，只要把变频器的内部电源不接入即可，只用一个电源，这是最佳办法。方案 2，可以使用双电源供电，改进后的电源如图 12-160 所示。

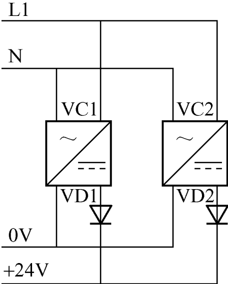



图 12-160　原理图

**【调试和故障诊断实例 4】**

故障现象：

某设备的电磁阀额定功率为 20 W，额定电压为 +24 V，电磁阀上电，系统工作正常，但断电时，会造成直流电源跳闸。

故障检查与排除：

这个跳闸应与短路无关。由于电磁阀的功率较大，断电后产生较大的感应电动势，对直流电源产生较大的干扰，因此直流电源跳闸。

解决方案 1：

加续流二极管即可解决，如图 12-161 所示，这是最容易想到的方法。

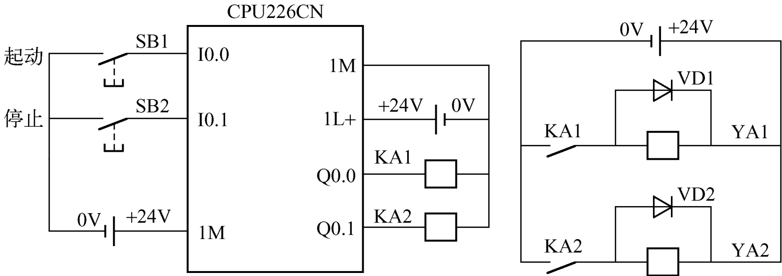



图 12-161　原理图

解决方案 2：

单独给冲击性负载供电。

**【调试和故障诊断实例 5】**

故障现象：

某设备的主回路有变频器，电源开关为剩余电断路器，运行时，经常跳闸。

故障检查与排除：

一般不能在变频器前使用剩余电断路器。因为变频器的大功率器件是 IGBT，容易造成剩余电断路器跳闸。通常使用一般断路器。

如果一定要使用剩余电断路器，则应选用剩余电较大的剩余电断路器。

**【调试和故障诊断实例 6】**

故障现象：

某纺织机械，配丹佛斯变频器，正常转速为 10000 r/min，正常起动和运行时，无故障，但停机阶段，数次烧坏变频器的电容。

故障检查与排除：

1）起动和运行阶段无故障表明变频器的功率足够。

2）设备的转速高，停机时巨大的动能转换成电能反冲到变频器的电容，过量的电量致使电容烧毁。

3）在直流回路中添加制动电阻后，问题解决，再无烧电容现象。

4）还有一种解决办法是在直流回路中加充放电电容，好处是在制动时存储电力，在运行时释放电力，可以节省能源。

**【调试和故障诊断实例 7】**

故障现象：

某人按如图 12-162 所示接线，之后学生按下 SB1、SB2 和 SB3 按钮，发现输入端的指示灯没有显示，PLC 中没有程序，但灯 HL 常亮，接线没有错误，+24 V 电源也正常。学生的分析是输入和输出接口烧毁，请问学生的分析是否正确。

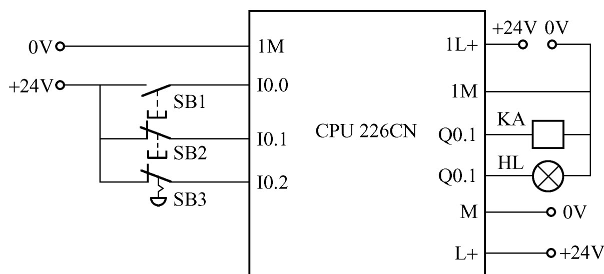



图 12-162　原理图

故障检查与排除：

1）实验设备的输入端口不会烧毁，因为输入接口电路有光电隔离电路保护，除非有较大电压（如交流 220 V）的误接入，而且烧毁输入接口一般也不会所有的接口同时烧毁。经过检查，发现接线端子 1 M 是“虚接”，压紧此接线端子后，输入端恢复正常。

2）错误接线容易造成晶体管输出回路的器件烧毁，晶体管的击穿会造成回路导通，从而造成 HL 灯常亮。

**【调试和故障诊断实例 8】**

故障现象：

某设备的控制系统由 PLC 和 5 台 G120 变频器组成，操作工反映每天晚上有 1～2 次变频器停机，上电后可正常使用，白天正常。

故障检查与排除：

维修检查发现：报警信息为“F06310”，晚上的电压偏高，为 420 V，而白天电压基本正常，为 390 V。晚上为过压报警。

安装了 20 kW 的自偶变压器，将电压调整到 370～400 V 后，问题得到解决。

**【调试和故障诊断实例 9】**

故障现象：

某设备的控制系统由 S7－300、数字量和模拟量模块组成，通电后发现模拟量模块的通道烧毁。(系统并无短路和过电压故障)

故障检查与排除：

维修检查发现：接线电工疏忽，将地线和零线接错，由于模拟量通道无光电隔离，所以容易烧毁。

将接错的线更正后，问题得到解决。

**【调试和故障诊断实例 10】**

故障现象：

某自动化设备上配有 PLC，最后一道工序是打标机，打标信号由 PLC 继电器的触点送给打标机，发现：90% 的情况打标机能正常工作，而 10% 不能打标。原理图如图 12－163 所示。

故障检查与排除：

经过检查，打标机无故障，自动化设备的前端也没有问题，分析原因是继电器触点闭合瞬间产生较大干扰，导致打标机不工作。加滤波后，工作正常，改进后的原理图如图 12－164 所示。

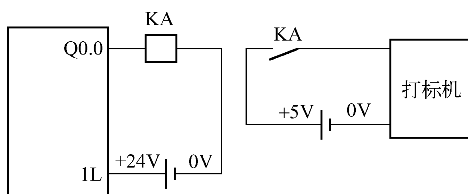



图 12-163　原理图

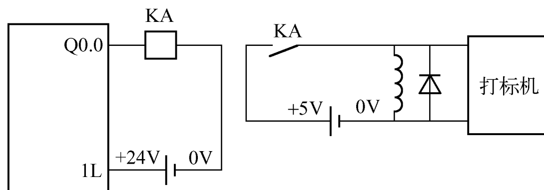



图 12-164　原理图

**【调试和故障诊断实例 11】**

故障现象：

某初学者，下载程序到 S7－300 后，SF 灯亮，但不知是硬件故障，还是程序错误，如何处理。

故障检查与排除：

删除下载的程序，只下载正确的硬件组态，检查 SF 指示灯是否亮，如不亮，则程序错误。

**【调试和故障诊断实例 12】**

故障现象：

某设备的控制器为 S7－300 和 CP343－1，发现 CPU 的 STOP 和 SF 灯亮，而 CP343－1 的灯都不亮。

故障检查与排除：

1）首先检查接线是否正确，有无断线、短路和接错线，DC 24 V 供电电压是否正常。

2）系统断电，检查模块是否安装牢固。

3）如排除以上问题，基本可以确定是 CP343－1 模块故障，需要拆下，更换新的模块；问题得以解决。

**【调试和故障诊断实例 13】**

故障现象：

如图 12-165 所示伺服系统，橙色为电源线，绿色为编码器线，使用时，偶尔有干扰现象。

故障检查与排除：

1）强电和弱电要分开，电源线和编码器线不能捆扎在一起。

2）电源线最好不要环绕，且间距大于 20 cm。

**【调试和故障诊断实例 14】**

故障现象：

维修人员发现 S7－200（继电器型输出）的 Q0. 3 烧毁，怎样解决？

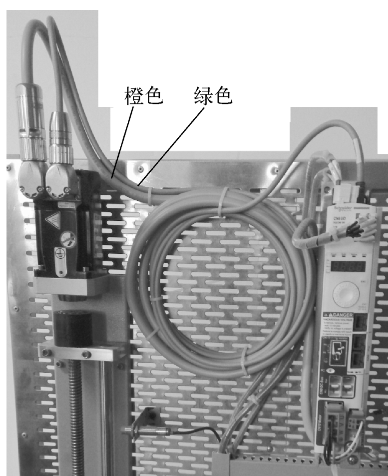



图 12-165　系统照片

故障检查与排除：

1）方案 1：拆开主板，更换继电器。这种方案对维修人员要求高。

2）方案 2：此 PLC 还有备用点 Q3.3，将 Q0.3 上的硬接线改到 Q3.3 上，再把程序中的 Q0.3 全部替换成 Q3.3，这种方案容易实现，但必须要源代码。

**【调试和故障诊断实例 15】**

故障现象：

某钢厂设备的控制系统由 1 台 CPU315－2DP、3 台 MM440 和两台 ET200M 以及 I/O 模块组成。其中一台变频器安装在一辆移动小车上。经常出现小车变频器接头松动而造成系统崩溃，生产厂家一直未能解决。请解决此问题。

故障检查与排除：

1）移动小车造成 PROFIBUS－DP 通信接头松动，这是很正常的事情，也很难防止其再次松动。

2）解决方案：小车上的变频器和 CPU 的信息沟通不采用 PROFIBUS－DP 通信，而直接用 ET200M 上的模拟量调速，起停等信息用 ET200M 上的 I/O 端子。

3）做此改进后，系统一直稳定运行。

**【调试和故障诊断实例 16】**

故障现象：

某设备的控制系统由 1 台 CPU315－2DP 和 10 个从站组成，采用 PROFIBUS－DP 通信。站间通信距离最远为 200 m。调试时，发现系统很不稳定，通信频繁出错。请查找原因。

故障检查与排除：

1）厂家的此设备已经出售多台，未发现类似原因，可排除设备设计和程序上的重大缺陷。

2）检查 DP 接头和所有屏蔽线是否接线都正常。

3）把波特率从 500 kbit/s 降至 9.6 kbit/s，也无明显改善。

4）检查现场，发现有多台大功率变频器和中频淬火炉，且频繁起停，后停止运行附近

所有设备，通信正常。故判断为干扰所致。

5）解决方案：将 DP 通信电缆改为光纤通信电缆后，问题得以解决。因为光纤抗干扰能力强。

**【调试和故障诊断实例 17】**

故障现象：

某设备的控制系统由 1 台 CPU315－2DP 和多个从站组成。有一台 AGV 运料小车与主站采用无线通信。试用一段时间后，发现系统不稳定，主要是对 AGV 小车控制不准确，有时甚至按下急停按钮也不停机。请查找原因。

故障检查与排除：

1）AGV 小车收不到急停信号，说明干扰信号比较大。对无线信号的影响比较大。这种影响有时甚至是致命的。

2）解决方案：将无线信号改为有线控制，控制线由软链拖拽，使用效果良好。

**【调试和故障诊断实例 18】**

故障现象：

S7－300 AI 模块，输入信号为 4～20 mA 电流信号，接线与程序均正确，但是在程序监控中显示接收的数据不确定，跳动很大，地线和屏蔽接线良好。

故障检查与排除：

1）方案 1：将信号负 AI－与 AI 模块的电源负短接，这种解决办法简单易行。

2）方案 2：在模拟量模块和传感器之间安装模拟信号隔离器。

**【调试和故障诊断实例 19】**

故障现象：

WinCC 通过 CP5621 通信卡与 PLC 进行 MPI 通信，所有组态、设置均正确，但是连接时通信时断时续，通过软件检测硬件，结果正常，但是通信仍一直存在此问题。

故障检查与排除：

将通信所用的 DP 电缆与 DP 接头，全部拆下，发现 DP 电缆的屏蔽层与 DP 接头内的接地没有连接，重新制作后再测试，通信正常。

**【调试和故障诊断实例 20】**

故障现象：

现场一台 CPU315－2DP 采用 PID 输出调节变频器的频率，发现 PID 输出到 100%（也就是 AO 模块输出最大 20 mA 电流信号）时，变频器的频率没有到最大的 50 Hz，检测电流只有 16 mA。

故障检查与排除：

查找问题，发现 AO 模块后面有隔离器，将隔离器拆除，AO 模块直接输出到变频器上，问题解决。

**【调试和故障诊断实例 21】**

故障现象：

检查时发现 CPU315－2DP 的信号线与动力线放在同一个桥架内，且现场设备信号没有传送到 PLC；所有接线正确。但是 PLC 的一个 DI 模块 LED 指示灯一直处于点亮状态。

故障检查与排除：

用万用表测量电压，竟然有 110 V。

原因：强电的感应电。

解决方案：强电和弱电布线分开。可见，强电和弱电分开布置是非常重要的。

**【调试和故障诊断实例 22】**

故障现象：

某钢厂改造一套输灰设备，原控制系统方案为：CPU414 带 ET200M 远程站，改造后，增加了硬件模块，所以程序必须要下载。改造时业主允许进行程序调试，造成的停机时间是半小时，问业主要求是否合理？

故障检查与排除：

先把组态和程序进行模拟仿真，确保程序正确。下载硬件组态 CPU 会停机重启，但下载软件 CPU 不会停机，因此，半小时停机时间是足够的，即使程序不正确，也可以重新下载，但必须保证硬件组态正确。

**【调试和故障诊断实例 23】**

故障现象：

某设备的控制系统由 1 台 CPU315 - 2DP、1 台 MM420（控制风机）、3 台软起动器（控制风机）和 3 台 ET200M 组成，ET200M 与 CPU315 - 2DP 为 PROFIBUS - DP 通信。MM420 模拟量频率给定，端子排起停控制，软起动器为直接端子排起停控制。

某日发现两台风机不能在远程中控室中用 HMI 控制运行，变频器控制不能就地起动，而软起动的可以就地起动。其他设备均可中控制室控制。

故障检查与排除：

在中控室发出软起动器起停信号，PLC 上有灯指示，表明信号到达 PLC，说明通信正常；接着查找，发现中间继电器不吸合，更换中间继电器即可。

在变频器处，用信号发生器给模拟量信号和端子排起停信号，也不能起动，表明变频器故障，需要维修。

**【调试和故障诊断实例 24】**

故障现象：

洗衣机不锈钢内筒生产线上的液压专机发生故障，油缸下行后无法上行，一直停留在下极限位置。手动进行上行操作也无效。

故障检查与排除：

此液压专机电气控制系统为 PLC 控制，检查液压系统，无故障。此油缸由两位五通控制电磁阀，直接用按压电磁阀换向阀芯，也无法换向。而此时发现电磁阀线圈有吸力。后测量发现电磁阀线圈有电压。但 PLC 信号输出指示灯显示无输出。再次检查发现 PLC 输出触点粘连，造成下降线圈一直得电。

故障排除：更换 PLC 输出触点，故障排除（此故障为电气系统设计不合理，PLC 输出点直接控制电磁阀线圈，而未经过中间继电器转换）。

**【调试和故障诊断实例 25】**

故障现象：

洗衣机装配生产线上的搬运机械手在工件到位后无夹紧动作，检测工件到位信号是对射式光传感器，根据 I/O 接线图，先检查 PLC 输入端信号是否正常，结果发现流水线有工件，

但输入端无信号。

故障检查与排除：

根据经验先重点检查红外线传感器及相关线路。经过现场测试工作正常；再检查相关信号线路没有发现断线等故障。短接信号输入端与公共端、PLC 输入端也没有信号，故障为 PLC 输入端内部光耦损坏。

更换到备用的 PLC 输入端后，故障排除。故障分析，此红外线传感器信号由继电器输出，通过接线端子送到控制柜，安装接线端子的接线盒的在设备的下面，由于接线盒锈蚀，线路凌乱，外部的潮气进入接线盒，致使信号端子附近强电窜入弱电回路，烧毁 PLC 光耦。

**【调试和故障诊断实例 26】**

故障现象：

洗衣机离合器生产线的轴承压入机在运行中突然发生故障，无自动运行，尝试手动操作进行复位也无动作。

故障检查与排除：首先检查电气控制系统，发现 PLC 面板上的电源指示灯闪烁。运行指示灯不亮，根据此现象推断是 PLC 故障，但更换 PLC 故障依旧。后查阅图纸发现，外接的两只 3 线式传感器使用的是 PLC 内部的 24 V 电源（厂家为节约成本）。后将 PLC 上 24 V 电源的正极接线断开，PLC 恢复正常。

经检查发现其中一只传感器的线缆破损，传感器的电源的正负极短路。从而造成 PLC 短路保护。经询问操作工人，故障发生前有一箱零件倾倒下来，故分析此故障的原因是倾倒下来的零件砸伤了传感器的线缆造成短路故障。

建议不要使用 PLC 的内部电源向传感器供电，而要使用外部开关电源供电。

**【调试和故障诊断实例 27】**

故障现象：

FR – E700 变频器刚起动时，E. 0C1 报警，请问可能是什么故障?

故障检查与排除：

1）先查手册，E. 0C1 为加速过流报警，最简单的方案是将加减速时间都延长，但仍然报警。

2）把电动机脱开，也就是不加负载，看有无报警，若无，则很可能是负载卡死或者过大，若仍然报警，则很可能变频器故障。

3）后发现是变频器故障，送厂家维修。

**【调试和故障诊断实例 28】**

故障现象：

某系统配有两套伺服系统。伺服系统拖动小车快速送料，调试时，发现振动较大，影响生产。

故障检查与排除：

1）先检查机械结构，对机械结构紧固后，振动有所改善。

2）后将伺服驱动器的加速和减速时间加大，问题最终解决。

**【调试和故障诊断实例 29】**

故障现象：

某设备的的运料翻斗，控制系统由 PLC 和变频器组成，翻斗从低处取料，再到高处，

再下移后卸料。运行时，变频器经常在下移过程中报警。系统有制动电阻。

故障检查与排除：

翻斗在下降过程中，翻斗带动电动机转动，电动机变成发电机，多余的电力使泵升电压升高，从而致使变频器过压报警。

将制动电阻的功率调大，问题得以解决；也可采用增加回馈单元或者直流共母线的方法加以解决。

**【调试和故障诊断实例 30】**

故障现象：

某设备的控制系统由 1 台 CPU314C－2DP 和两个从站（3 号和 4 号从站）组成，采用 PROFIBUS－DP 通信。某日早上设备不能正常开机，发现 CPU 的 SF 和 BF 灯亮。请进行故障诊断。

故障检查与排除：

1）首先判断通信故障，采用诊断视图诊断，如图 12-166 所示。

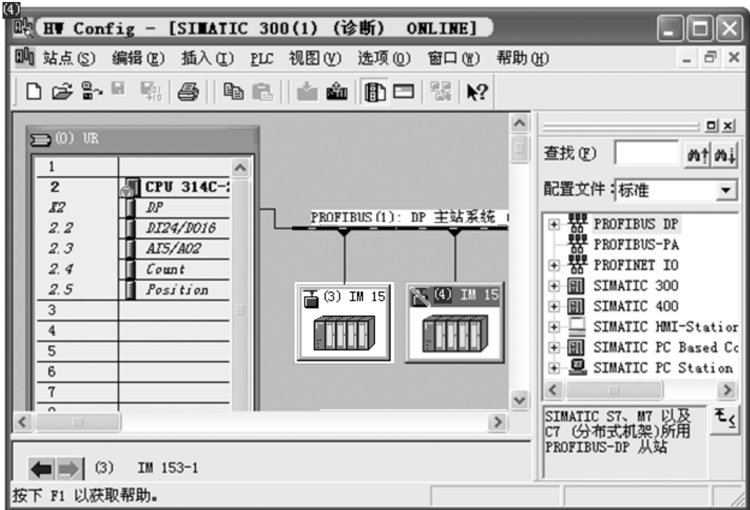



图 12-166　硬件组态

2）仔细检查后发现，第 3 站 DP 接头上的终端电阻置于“ON”，将其拨到“OFF”即可。

# 第 13 章　S7 - 300/400 PLC 工程应用

本章是前面章节内容的综合应用，将介绍两个典型的 PLC 控制系统集成过程的案例，供读者模仿学习。

## 13.1　啤酒灌装线系统的 PLC 控制

**【例 13-1】** 有两条啤酒生产线，由一台 S7 - 300 控制。当啤酒线起动运行啤酒瓶到位后，传送带停止转动，开始灌装啤酒，装满后，传送带转动；当啤酒瓶到达灌装线尾部时，系统自动计数，并显示。1 号线示意图如图 13-1 所示。

2 号线示意图如图 13-2 所示。

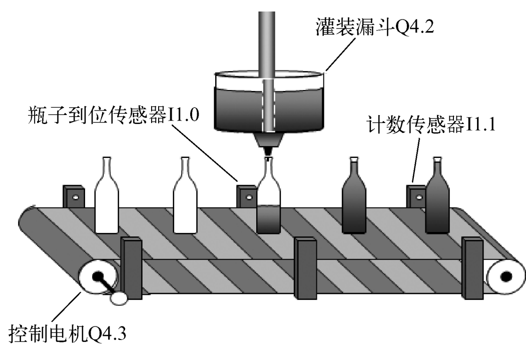



图 13-1　1 号线示意图

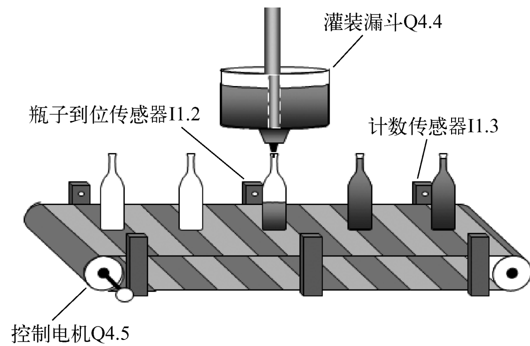



图 13-2　2 号线示意图

面板示意图如图 13-3 所示。请设计此系统，并编写梯形图程序。

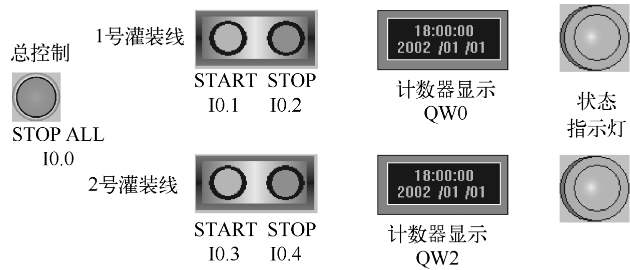



图 13-3　面板示意图

### 13.1.1　系统软硬件配置

**1. 系统的软硬件**

① 1 套 STEP 7 V5. 5 SP4。

② 1 台 CPU 314C - 2DP。

③ 1 根编程电缆。

④ 两台 SM322。

**2. PLC 的 I/O 分配**

PLC 的 I/O 分配见表 13-1。

**表 13-1 PLC 的 I/O 分配表**

| 名　　称 | 符　　号 | 输　入　点 | 名　　称 | 符　　号 | 输　出　点 |
|---|---|---|---|---|---|
| 总控制按钮 | SB1 | I0.0 | 1 号线计数 | | QW0 |
| 1 号线起动按钮 | SB2 | I0.1 | 2 号线计数 | | QW2 |
| 1 号线停止按钮 | SB3 | I0.2 | 1 号线显示 | HL1 | Q4.0 |
| 2 号线起动按钮 | SB4 | I0.3 | 2 号线显示 | HL2 | Q4.1 |
| 2 号线停止按钮 | SB5 | I0.4 | 1 号线电动机 | KA1 | Q4.3 |
| 1 号线瓶子到位传感器 | SQ1 | I1.0 | 2 号线电动机 | KA2 | Q4.5 |
| 1 号线计数传感器 | SQ2 | I1.1 | 1 号线电磁阀 | YA1 | Q4.2 |
| 2 号线瓶子到位传感器 | SQ3 | I1.2 | 2 号线电磁阀 | YA2 | Q4.4 |
| 2 号线计数传感器 | SQ4 | I1.3 | | | |

**3. 控制系统的接线**

控制系统的接线如图 13-4。

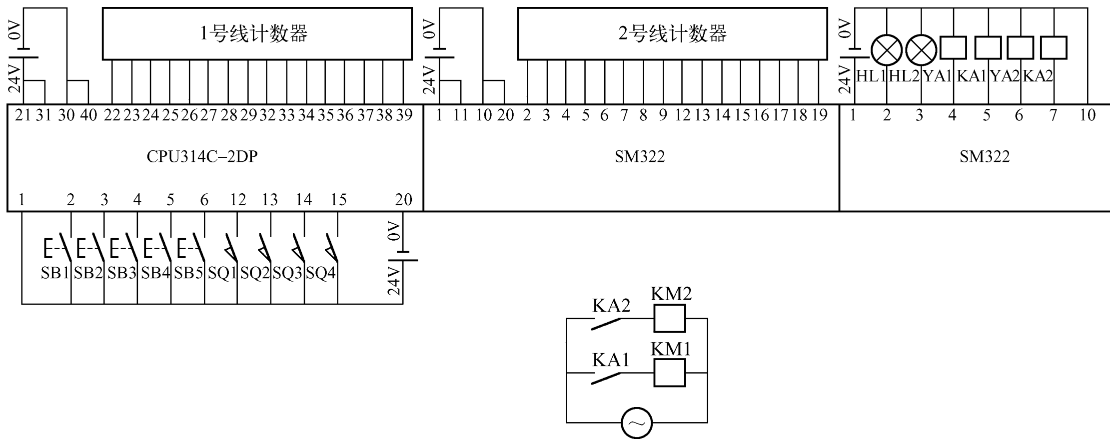



图 13-4　接线图

**4. 硬件组态**

新建项目，命名为"灌装线"，配置 1 块模块 CPU314C－2DP 和两块 DO16x24V，如图 13-5所示，并将 CPU314C－2DP 的输入地址修改为"IB0～IB2"，输出地址修改为"QB0～QB1"；然后将第一块 DO16x24V 的输出地址修改为"QB2～QB3"；将第二块 DO16x24V 的输出地址修改为"QB4～QB5"；这些地址在编写程序时，必须与之一一对应。

## 13.1.2　编写程序

**1. 输入符号表**

如图 13-6 所示输入符号表。在实际工程中，这项工作不能省略。

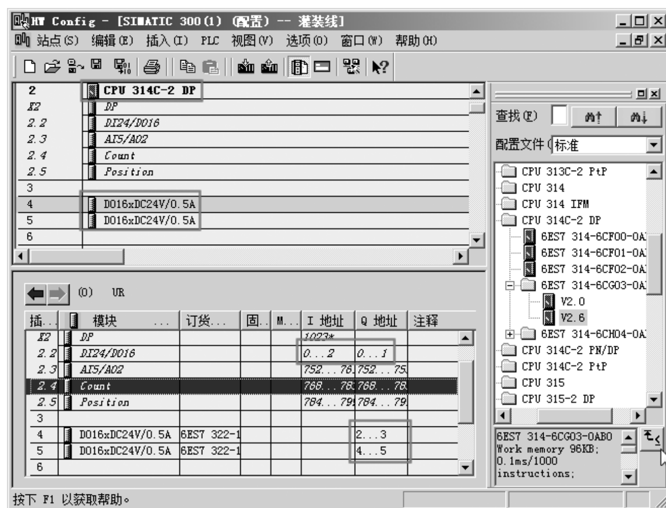



图 13-5　硬件组态

符号编辑器 - [S7 程序(1) (符号) -- 灌装线\SIMATIC 300(1)\CPU 314C-2 DP]

| | 状态 | 符号 | 地址 | | 数据类型 | 注释 |
|---|---|---|---|---|---|---|
| 1 | | Bottling_Control | FB | 1 | FB 1 | |
| 2 | | Cntr_1 | C | 1 | COUNTER | Line1 Bottling Counter |
| 3 | | Cntr_2 | C | 2 | COUNTER | Line2 Bottling Counter |
| 4 | | Display_1 | QW | 0 | WORD | Line1 Count Display |
| 5 | | Display_2 | QW | 2 | WORD | Line2 Count Display |
| 6 | | FB1 | DB | 1 | FB 1 | |
| 7 | | Fill_1 | Q | 4.2 | BOOL | Line1 Bottling Control |
| 8 | | Fill_2 | Q | 4.4 | BOOL | Line2 Bottling Control |
| 9 | | Filling_Status_1 | M | 0.0 | BOOL | Line1 Bottling Status |
| 10 | | Filling_Status_2 | M | 0.1 | BOOL | Line2 Bottling Status |
| 11 | | Line1_Status | Q | 4.0 | BOOL | Line1 Status(Run/Stop) |
| 12 | | Line2_Status | Q | 4.1 | BOOL | Line2 Status(Run/Stop) |
| 13 | | Motor_1 | Q | 4.3 | BOOL | Line1 Motor Control |
| 14 | | Motor_2 | Q | 4.5 | BOOL | Line2 Motor Control |
| 15 | | Operation_Con... | FC | 1 | FC 1 | |
| 16 | | Sensor_Fill_1 | I | 1.0 | BOOL | Line1 Bottling Sensor |
| 17 | | Sensor_Fill_2 | I | 1.2 | BOOL | Line2 Bottling Sensor |
| 18 | | Sensor_Full_1 | I | 1.1 | BOOL | Line1 Bottling Full Sensor |
| 19 | | Sensor_Full_2 | I | 1.3 | BOOL | Line2 Bottling Full Sensor |
| 20 | | Start_L1 | I | 0.1 | BOOL | Line1 Start Signal |
| 21 | | Start_L2 | I | 0.3 | BOOL | Line2 Start Signal |
| 22 | | Stop_All | I | 0.0 | BOOL | Stop All Lines |
| 23 | | Stop_L1 | I | 0.2 | BOOL | Line1 Stop Signal |
| 24 | | Stop_L2 | I | 0.4 | BOOL | Line2 Stop Signal |
| 25 | | Timer_1 | T | 1 | TIMER | Line1 Bottling Timer |
| 26 | | Timer_2 | T | 2 | TIMER | Line2 Bottling Timer |
| 27 | | | | | | |

按下 F1 获取帮助。

图 13-6　符号表

## 2. 编写 FC1 的梯形图

1）新建功能 FC1，再在程序编辑器中声明 3 个输入参数：Start、Stop 和 Stop_All，如图 13-7所示。

Interface
IN
Start
Stop
Stop_All
OUT
Plant_On
IN_OUT

Contents Of: 'Environment\Interface\IN'

| Name | Data Type | Comment |
|---|---|---|
| Start | Bool | 启动信号 |
| Stop | Bool | 停止信号 |
| Stop_All | Bool | 总停信号 |

图 13-7　声明输入参数

2）在程序编辑器中声明 1 个输出参数：Plant_On，如图 13-8 所示。

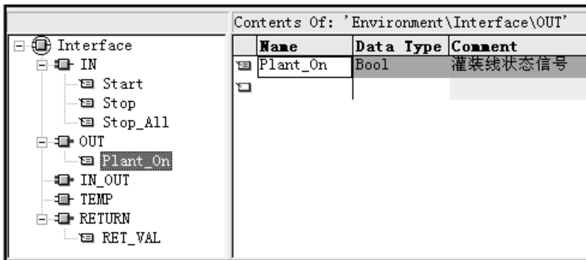



图 13-8　声明输出参数

3）编写 FC1 的梯形图，如图 13-9 所示，其功能实际就是起停控制。

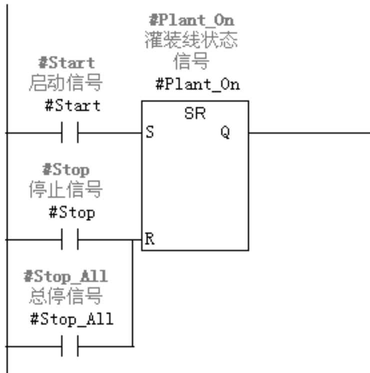



图 13-9　FC1 中的梯形图

**3. 编写 FB1 的梯形图**

1）新建功能块 FB1，再在程序编辑器中声明 5 个输入变量，要特别注意变量的数据类型。如图 13-10 所示。

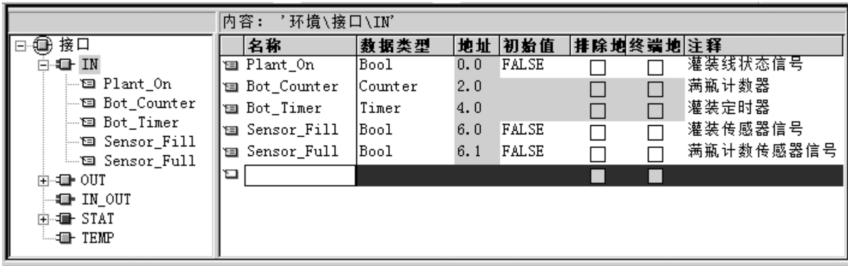



图 13-10　声明输入参数

2）在程序编辑器中声明两个输出变量，如图 13-11 所示。

3）编写 FB1 的梯形图，如图 13-12 所示。

**4. 编写 OB1 的梯形图**

主程序梯形图如图 13-13 所示。

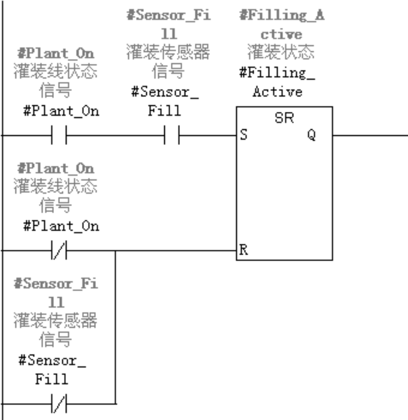
内容：'环境\接口\OUT'

- 接口
  - IN
  - OUT
    - Display
    - Filling_Active
  - IN_OUT
  - STAT
  - TEMP

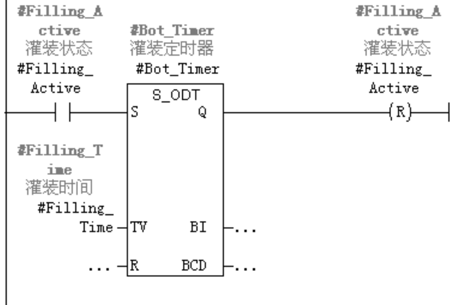
| 名称 | 数据类型 | 地址 | 初始值 | 排除 | 终端地址 | 注释 |
|---|---|---|---|---|---|---|
| Display | Word | 8.0 | W#16#0 | ☐ | ☐ | BCD 码计数值 |
| Filling_Active | Bool | 10.0 | FALSE | ☐ | ☐ | 灌装状态 |
| | | | | ☐ | ☐ | |

图 13-11　声明输出参数

**程序段 1：标题：**




**程序段 2：标题：**




**程序段 3：标题：**

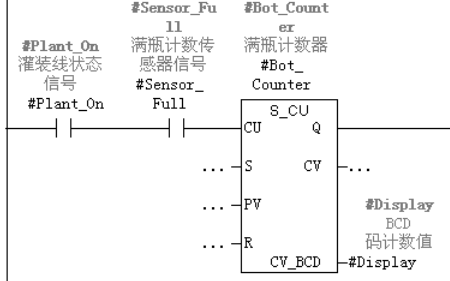
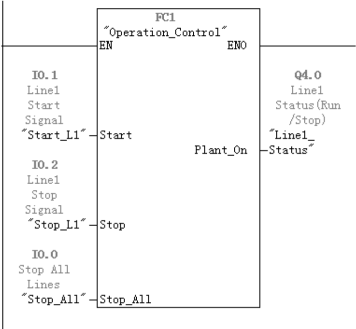



图 13-12　FB1 中的梯形图

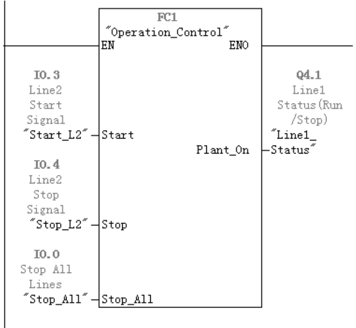
**程序段 1**：标题：

1号灌装线的启停控制：将控制面板的输入信号作为FC1的输入参数，FC1的输出参数为1号线运行状态




**程序段 2**：标题：

2号灌装线的启停控制。




**程序段 3**：标题：

1号灌装线的运行状态信号（Line_1_Status），计数器Cntr_1、定时器Time_1和1号灌装线的传感器信号，FB1的满瓶计数值输出到Display_1，输出灌装状态信号Filling_status_1

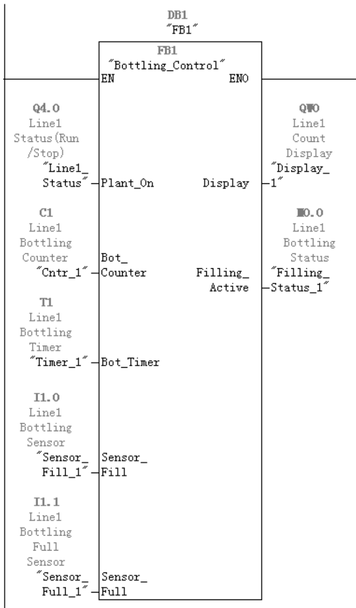
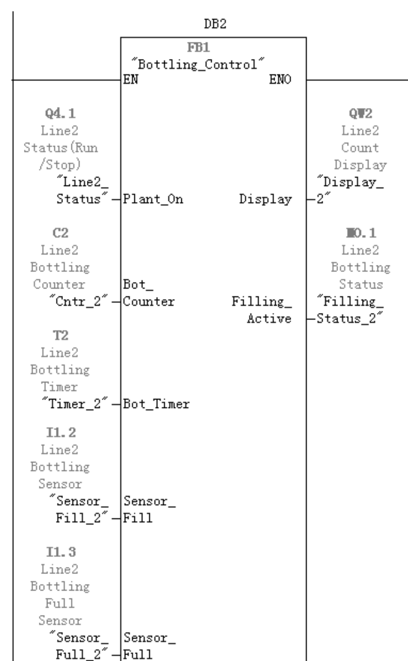



图 13-13　OB1 中的梯形图

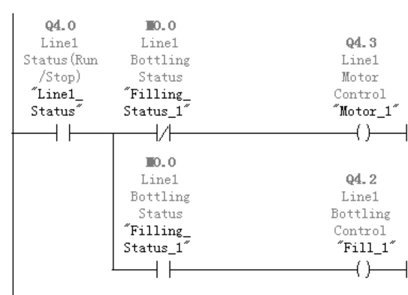
**程序段 5：**标题：

2号灌装线的运行状态信号




**程序段 5：** Line1 Motor Control

1号灌装线动作




**程序段 6：** Line1 Motor Control

2号灌装线动作

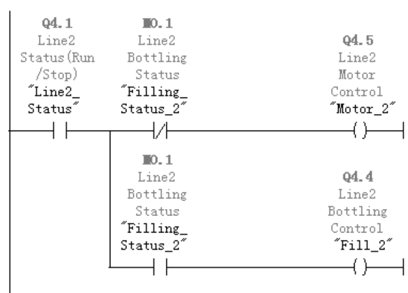



图 13-13　OB1 中的梯形图（续）

## 13.2 干粉喷射系统的 PLC 控制

**【例 13-2】** 在烟气净化干法脱酸系统中，干粉喷射系统由干粉分配器、干粉给料器及干粉喷射风机组成。其中干粉分配器将干粉仓中物料均匀分配给干粉给料器，控制器将据烟气实际值控制干粉给料器运行速度，要求干粉给料器速度在一定的范围内无级可调。当物料过多或者卡死设备时，系统能及时保护。干粉喷射风机将干粉输送至烟道，与酸性气体反应。为了防止管道堵塞，输送管道上配置压力检测装置，当检测到管道压力值超过一定值后，系统停止给料并发出报警。系统现场设备通过 PROFIBUS - DP 通信将采集到的现场参数传送至 DCS 控制室，同时从 DCS 的控制参数通过 PROFIBUS - DP 通信传送到系统的现场设备。重要的参数要求就地显示并方便修改。

本系统有两套设备，互为备用。干粉分配器电动机功率为 1.5 kW，干粉给料器电机功率为 1.1 kW，干粉喷射风机电机功率为 3 kW。系统运行时，当检测到管道压力值超过 30 kPa，（此数值可根据实际情况调整）视作管路堵塞，系统停止运行并发出警报，若检测到压力值低于 5 kPa，视作系统管道里无物料，也要求报警提示。本系统最大干粉喷射量为 300 kg/h。DCS 回传的烟气酸性气体指标为 $SO_2$ 和 HCL 气体的浓度，其中 $SO_2$ 的浓度最大为 300 $mg/Nm^3$，HCL 的浓度最大为 1800 $mg/Nm^3$，$SO_2$ 和 HCL 的浓度测试在其他系统完成，数据参数由 DCS 采集，本设计中要用到这些参数。

### 13.2.1 硬件系统集成

**1. 分析问题**

根据已知的工艺要求，分析结论如下：

1）干粉给料器电动机的速度要求无级可调，所以干粉给料器电动机由变频器拖动。

2）当设备卡死时，系统能及时保护。当载荷超过一定数值时（特别是电动机卡死时），电流急剧上升，当电流达到一定数值时，相关的电气器件会给出报警信号。

3）由于本系统并不复杂，所以常规 PLC 即可满足要求，考虑到整个系统采用西门子系列 PLC 控制，因此该系统的 PLC 选定为 S7 - 300 PLC。

4）干粉喷射管路的堵塞以压力为判断依据，当压力超过 30 kPa 值时，视为管路堵塞；当压力低于一定的压力时，视为缺料。压力测量选用压力传感器。因此，PLC 系统要配置模拟量输入模块。

5）要求系统参数和压力数值能显示，所以需要配置触摸屏或者其他显示设备。

6）与系统相关的监控参数都要传送至 DCS 控制室因此需要通信，由于选用的是 S7 - 300 系列 PLC，因此选用 PROFIBUS - DP 通信协议。

7）系统相关控制量的数据如下：

干粉喷射量：0 ~ 300 Kg/h。

变频器频率：0 ~ 50 Hz。

$SO_2$ 的浓度：0 ~ 300 $mg/Nm^3$。

HCL 的浓度：0 ~ 1800 $mg/Nm^3$。

有两种酸性气体，且都需要处理达标。

$SO_2$ 处理后的目标浓度值：0 ~ 50 mg/Nm$^3$。

HCL 处理后的目标浓度值：0 ~ 10 mg/Nm$^3$。

**2. 主要软硬件**

① 1 套 STEP 7 V5.5 SP4。

② 1 台 CPU 315 -2DP。

③ 1 台 SM321、SM322、SM331 和 SM332。

④ 两台 MM440 变频器。

⑤ 1 根编程电缆。

系统的硬件组态如图 13-14 所示。

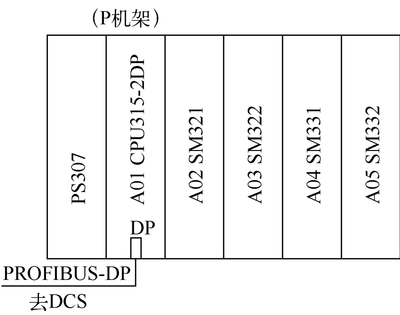



图 13-14　系统硬件组态图

**3. PLC 的 I/O 分配**

PLC 的 I/O 分配见表 13-2。

**表 13-2　PLC 的 I/O 分配表**

| 名　　称 | 符　　号 | 输　入　点 | 名　　称 | 符　　号 | 输　出　点 |
|---|---|---|---|---|---|
| 干粉分配器远程 | SA1 | I0.0 | 干粉分配器释放 | YA1 | Q0.0 |
| 干粉分配器运行 | SB1 | I0.1 | 1#干粉给料器释放 | KA2 | Q0.1 |
| 干粉分配器故障 | QF1 | I0.2 | 2#干粉给料器释放 | KA3 | Q0.2 |
| 1#干粉给料器远程 | SA2 | I0.3 | 1#干粉喷射风机释放 | KM1 | Q0.3 |
| 1#干粉给料器运行 | SB2 | I0.4 | 2#干粉喷射风机释放 | KM2 | Q0.4 |
| 1#干粉给料器故障 | QF4 | I0.5 | 系统故障警报 | HL1 | Q0.5 |
| 2#干粉给料器远程 | SA3 | I0.6 | 1#干粉给料器频率设定 | | PQW200 |
| 2#干粉给料器运行 | SB3 | I0.7 | 2#干粉给料器频率设定 | | PQW202 |
| 2#干粉给料器故障 | QF5 | I1.0 | | | |
| 1#干粉喷射风机远程 | SA4 | I1.1 | | | |
| 1#干粉喷射风机运行 | SB4 | I1.2 | | | |
| 1#干粉喷射风机故障 | QF2 | I1.3 | | | |
| 2#干粉喷射风机远程 | SA5 | I1.4 | | | |
| 2#干粉喷射风机运行 | SB5 | I1.5 | | | |
| 2#干粉喷射风机故障 | QF3 | I1.6 | | | |
| 干粉仓低料位 | SQ1 | I1.7 | | | |
| 系统急停按钮 | SB6 | I2.0 | | | |
| 系统复位按钮 | SB7 | I2.1 | | | |
| 1#干粉给料器频率检测 | | PIW200 | | | |
| 2#干粉给料器频率检测 | | PIW202 | | | |
| 1#干粉管路压力变送器 | | PIW204 | | | |
| 2#干粉管路压力变送器 | | PIW206 | | | |

**4. 控制系统原理图**

控制系统的主回路原理图如图 13-15 所示。注意 KA7 和 KA9 的线圈如图 13-16 所示，频率反馈信号 501、502、503 和 504 来自图 13-20，频率给定信号 401、402、403 和 404 与图 13-19 对应端子关联。当 1#干粉给料器的变频器故障时，MM4 -1 和 MM4 -2 触头闭合，使得如图 13-16 的 KA8 线圈得电，KA8 的常闭触头断开，从而导致 KA7 线圈断电，1#干粉

给料器的变频器停机。同理，2#干粉给料器的变频器故障时，也会导致2#干粉给料器的变频器停机，起到保护作用。

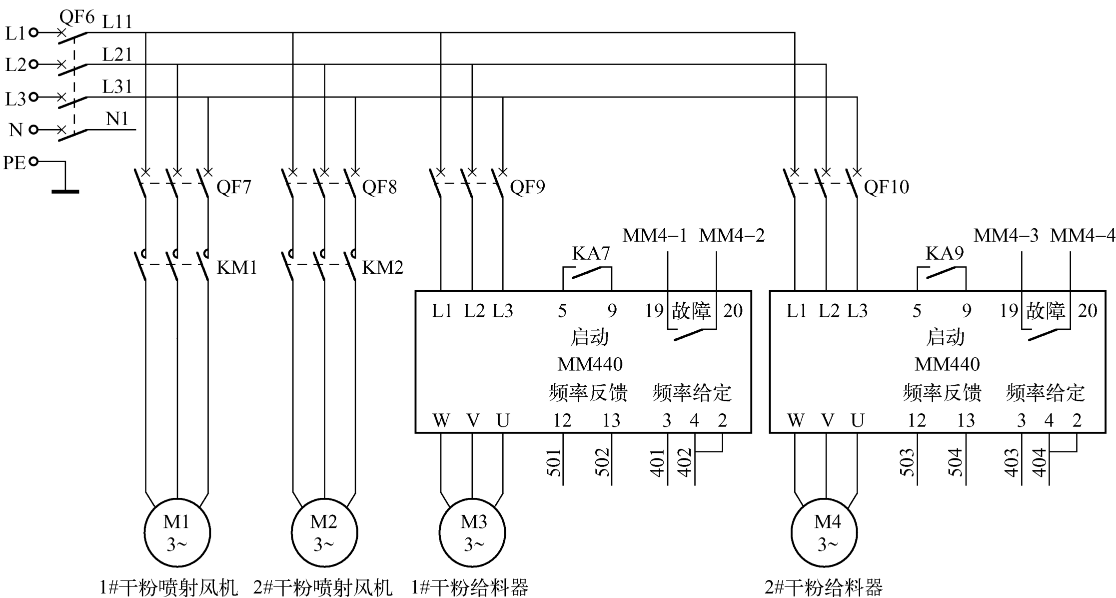


图 13-15 主回路

控制系统的控制回路需要分别对1#干粉给料器、2#干粉给料器、1#干粉喷射风机和2#干粉喷射风机进行控制，由于控制原理类似，限于篇幅，这里仅设计了1#干粉给料器的控制原理图，如图13-16所示。

本系统的控制分为“远程”和“就地”两种模式。所谓“就地”模式控制，就是在工作现场对设备进行控制，而“远程”模式是在控制室对现场设备进行控制，这在工程中较为常用。当“就地”模式时，SA6旋转到“就地”模式，当按下SB9按钮时，KA7的线圈得电，自锁，使得图13-15中的KA7常开触头闭合，从而使得1#干粉给料器的电动机运行。当“远程”模式时，SA1旋转到“远程”模式，当按下HMI中的压下起动按钮时，PLC的输出模块使KA2的线圈得电，自锁，使得图13-16中的KA2常开触头闭合，从而使得1#干粉给料器的电动机运行。

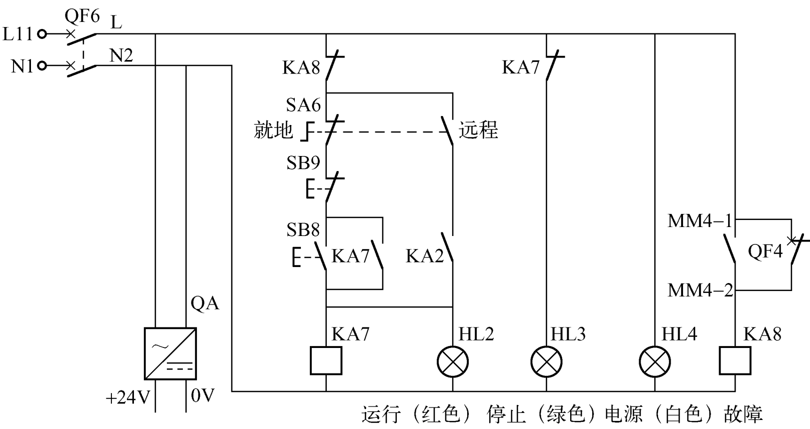


图 13-16 控制回路

控制柜的数字量输入模块原理图如图 13-17 所示。

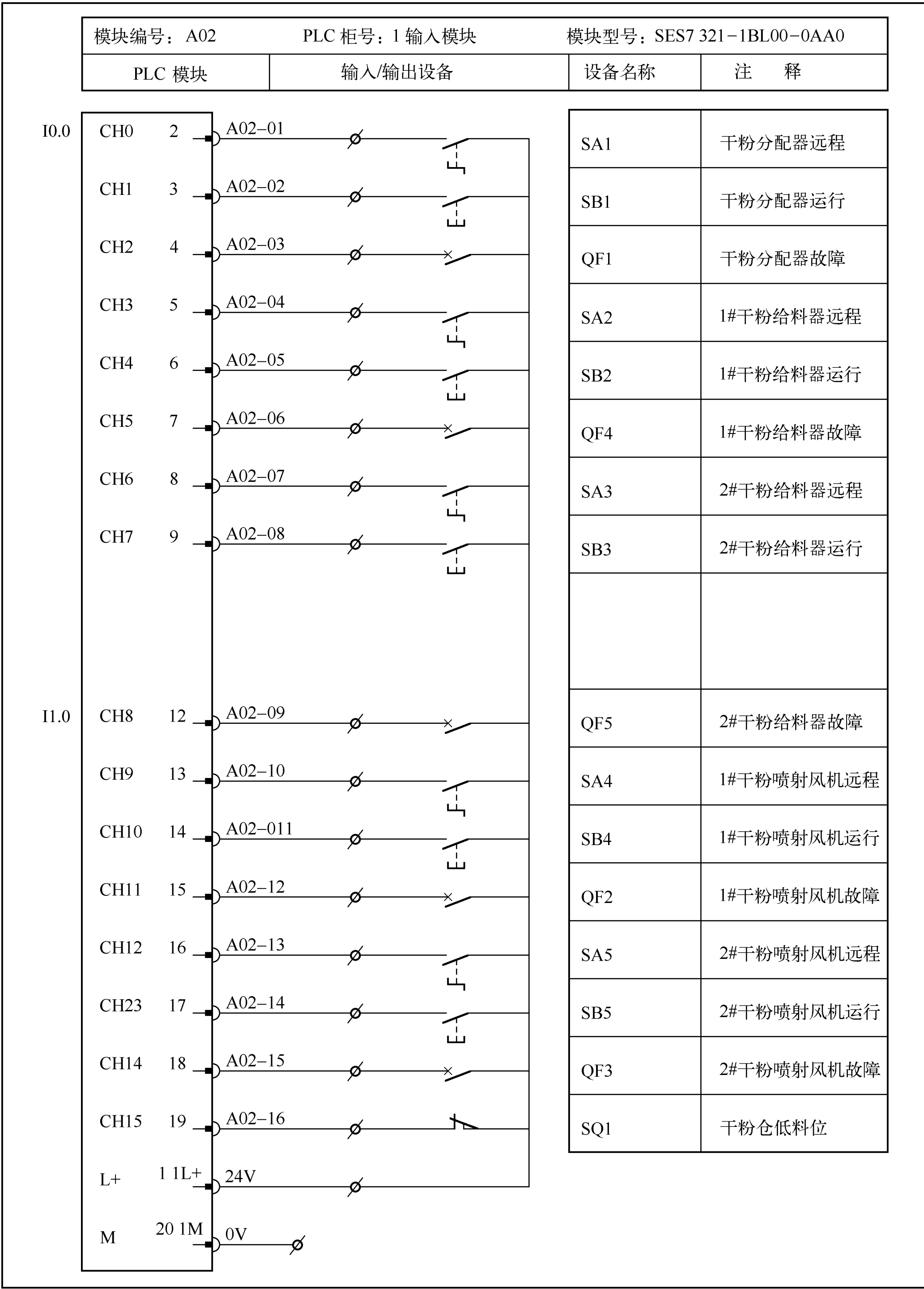



图 13-17 数字量输入模块原理图

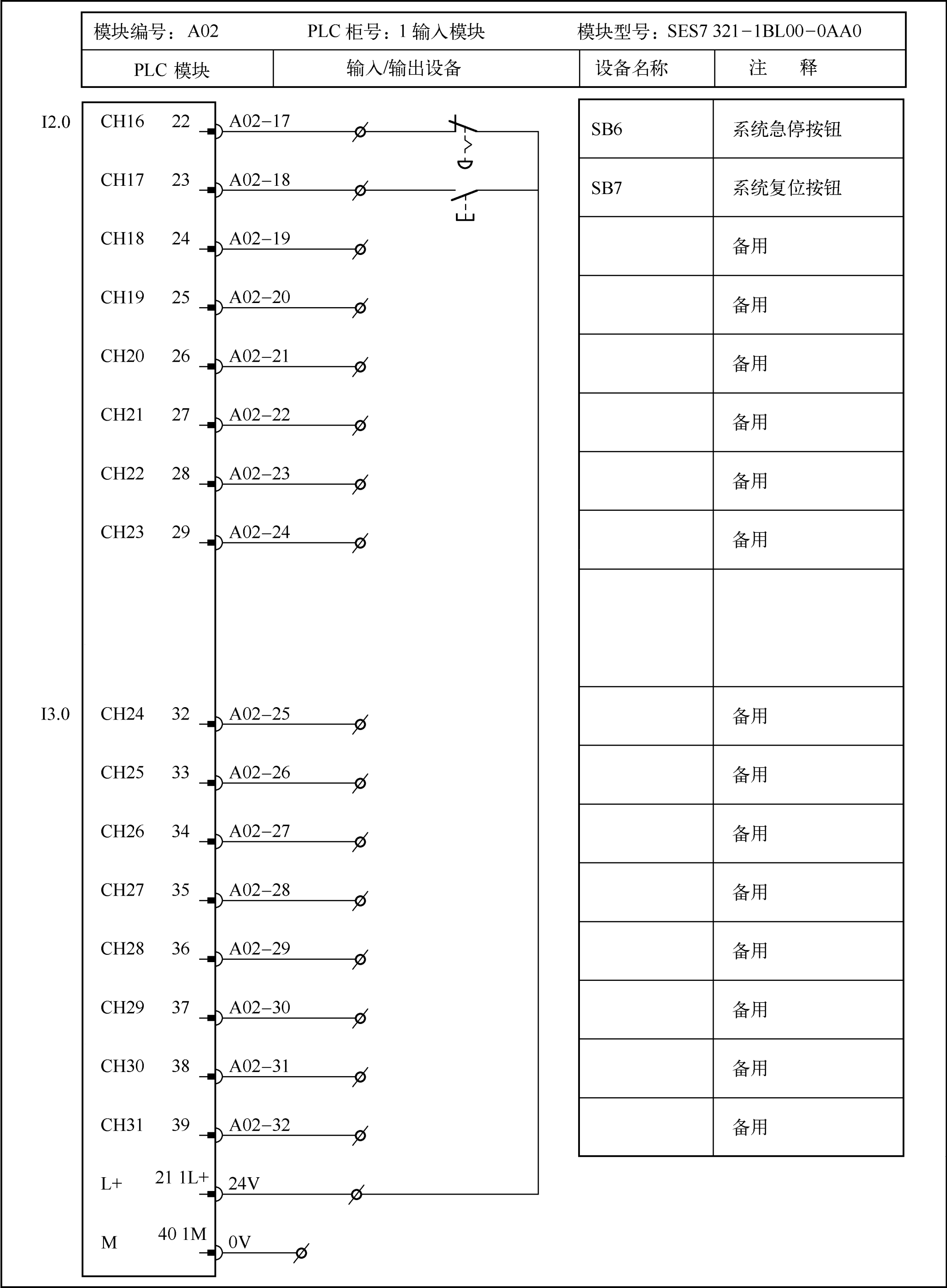



图 13-17　数字量输入模块原理图（续）

控制柜的数字量输出模块原理图如图 13-18 所示。

控制柜的模拟量输入模块原理图如图 13-19 所示。

控制柜的模拟量输出模块原理图如图 13-20 所示。

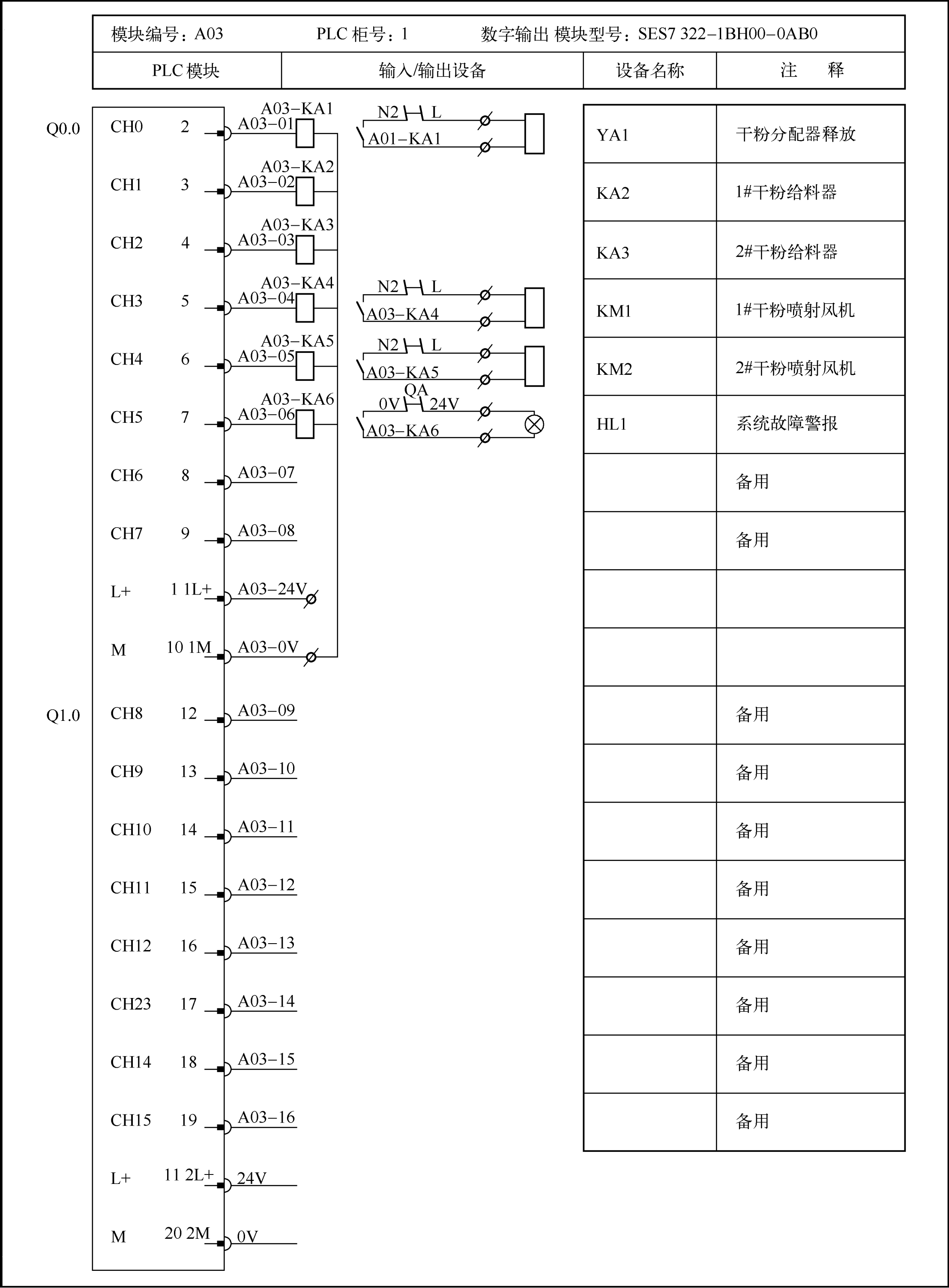



图 13-18　数字量输出模块原理图

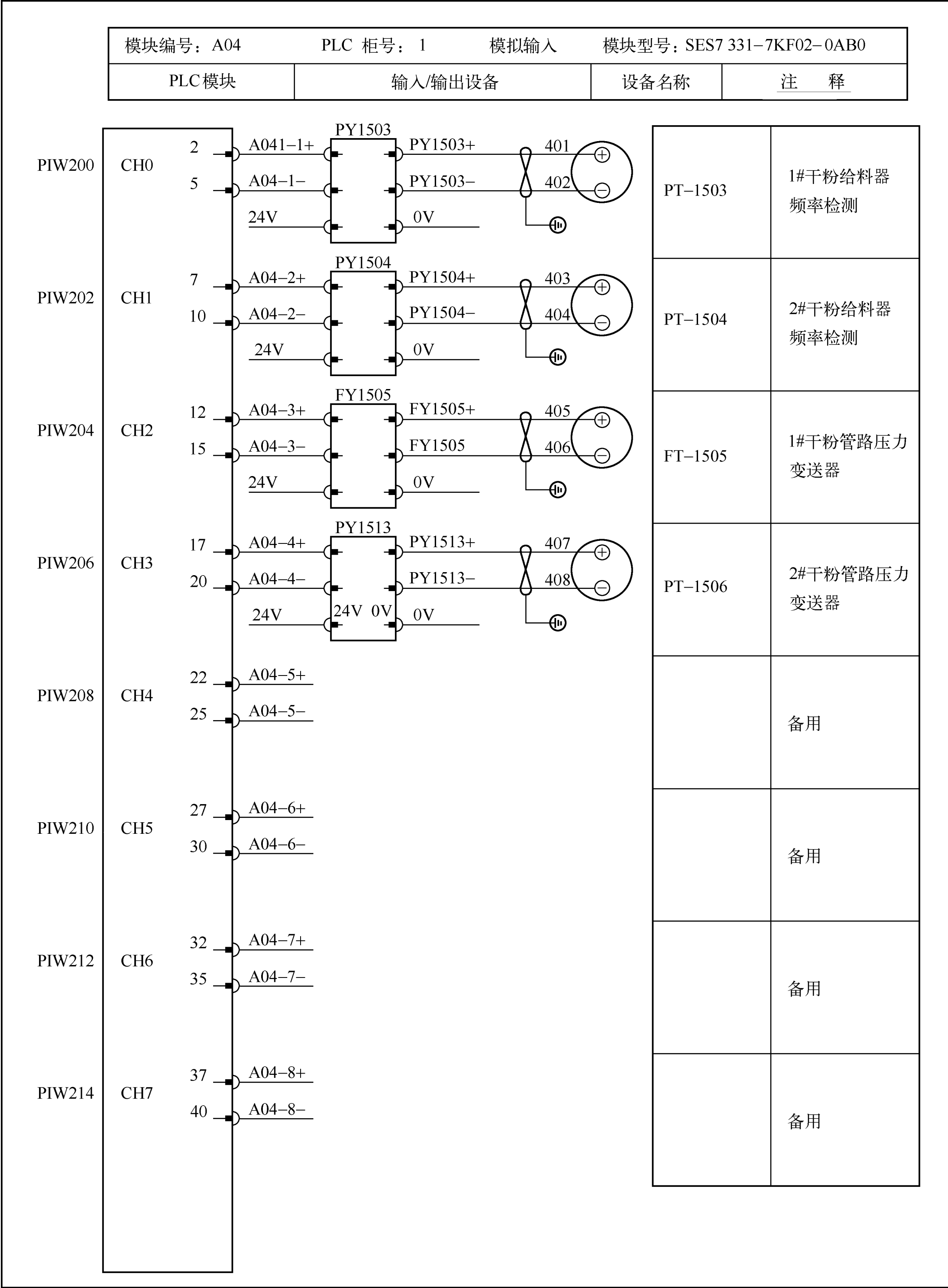



图 13-19　模拟量输入模块原理图

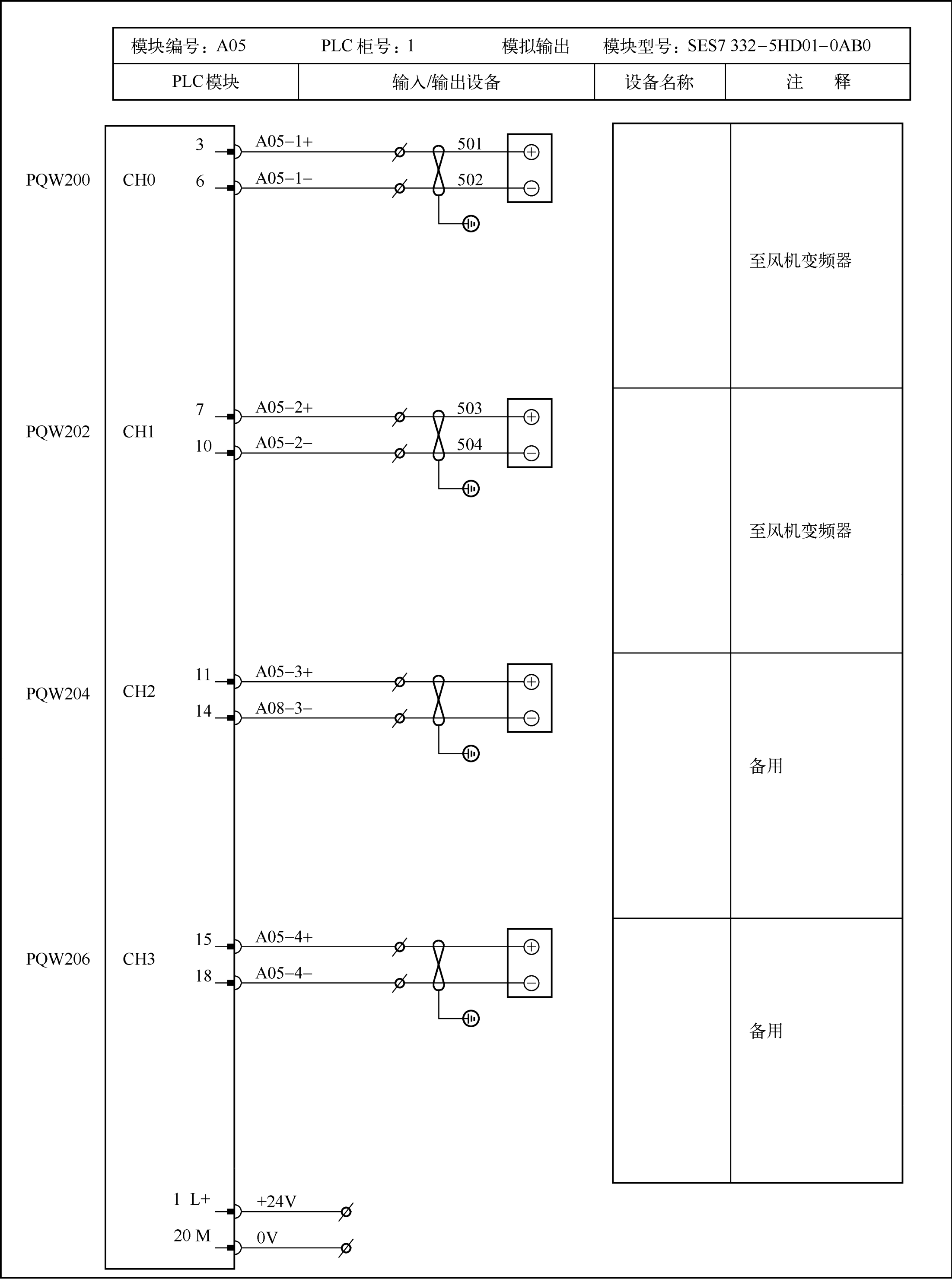



图 13-20　模拟量输出模块原理图

## 13.2.2 编写程序

### 1. 输入符号表

符号表见表 13-3。

表 13-3　符号表

| 符　　号 | 地　　址 | 数 据 类 型 | 注　　释 |
|---|---|---|---|
| D_Motor | UDT　1 | UDT　1 | |
| D_Motor_SUB | FB　11 | FB　11 | |
| DCS 读数据 | DB　1 | DB　1 | |
| DCS 读写数据 | FC　10 | FC　10 | |
| DCS 写数据 | DB　2 | DB　2 | |
| I000 | I　0.0 | BOOL | 干粉分配器远程/就地 |
| I001 | I　0.1 | BOOL | 干粉分配器运行 |
| I002 | I　0.2 | BOOL | 干粉分配器故障 |
| I003 | I　0.3 | BOOL | 1#干粉给料器远程/就地 |
| I004 | I　0.4 | BOOL | 1#干粉给料器运行 |
| I005 | I　0.5 | BOOL | 1#干粉给料器故障 |
| I006 | I　0.6 | BOOL | 2#干粉给料器远程/就地 |
| I007 | I　0.7 | BOOL | 2#干粉给料器运行 |
| I010 | I　1.0 | BOOL | 2#干粉给料器故障 |
| I011 | I　1.1 | BOOL | 1#干粉喷射风机远程/就地 |
| I012 | I　1.2 | BOOL | 1#干粉喷射风机运行 |
| I013 | I　1.3 | BOOL | 1#干粉喷射风机故障 |
| I014 | I　1.4 | BOOL | 2#干粉喷射风机远程/就地 |
| I015 | I　1.5 | BOOL | 2#干粉喷射风机运行 |
| I016 | I　1.6 | BOOL | 2#干粉喷射风机故障 |
| I017 | I　1.7 | BOOL | 干粉仓低料位低 |
| I020 | I　2.0 | BOOL | 系统急停按钮 |
| I021 | I　2.1 | BOOL | 系统复位按钮 |
| Int_TO_Rell SUB | FC　21 | FC　21 | 十六位整型数据转换为浮点数类型 |
| Int_TO_Time SUB | FC　23 | FC　23 | 十六位整型数据转换为时间类型 |
| M000 | M　0.0 | BOOL | SIM |
| M001 | M　0.1 | BOOL | SIMFault |
| M002 | M　0.2 | BOOL | BlockOut |
| M003 | M　0.3 | BOOL | 常开点 |
| M004 | M　0.4 | BOOL | 常闭点 |
| M010 | M　1.0 | BOOL | DCS 控制按钮 |
| M012 | M　1.2 | BOOL | 系统选择 |
| M100 | M　10.0 | BOOL | 干粉分配器故障 |
| M101 | M　10.1 | BOOL | 干粉分配器超时故障 |
| M102 | M　10.2 | BOOL | 1#干粉给料器故障 |
| M103 | M　10.3 | BOOL | 1#干粉给料器超时故障 |
| M104 | M　10.4 | BOOL | 2#干粉给料器故障 |
| M105 | M　10.5 | BOOL | 2#干粉给料器超时故障 |
| M106 | M　10.6 | BOOL | 1#干粉喷射风机故障 |
| M107 | M　10.7 | BOOL | 1#干粉喷射风机超时故障 |

（续）

| 符　　号 | 地　　址 | 数据类型 | 注　　释 |
| --- | --- | --- | --- |
| M110 | M　　11.0 | BOOL | 2#干粉喷射风机故障 |
| M111 | M　　11.1 | BOOL | 2#干粉喷射风机超时故障 |
| M112 | M　　11.2 | BOOL | 干粉仓低料位报警 |
| PID 调节 | FB　　3 | FB　　3 | |
| PIW200 | PIW　200 | INT | 1#干粉给料器频率检测 |
| PIW202 | PIW　202 | INT | 2#干粉给料器频率检测 |
| PIW204 | PIW　204 | INT | 1#干粉管路压力变送器 |
| PIW206 | PIW　206 | INT | 2#干粉管路压力变送器 |
| PQW200 | PQW　200 | WORD | 1#干粉给料器频率设定 |
| PQW202 | PQW　202 | WORD | 2#干粉给料器频率设定 |
| Q000 | Q　　0.0 | BOOL | 干粉分配器释放 |
| Q001 | Q　　0.1 | BOOL | 1#干粉给料器 |
| Q002 | Q　　0.2 | BOOL | 2#干粉给料器 |
| Q003 | Q　　0.3 | BOOL | 1#干粉喷射风机 |
| Q004 | Q　　0.4 | BOOL | 2#干粉喷射风机 |
| Q005 | Q　　0.5 | BOOL | 系统故障警报 |
| Real_TO_Int SUB | FC　　22 | FC　　22 | |
| RTD_TO_Real SUB | FC　　20 | FC　　20 | |
| 控制数据 | DB　　31 | DB　　31 | |
| 系统设备 | FB　　1 | FB　　1 | |
| 压力数据 | FB　　10 | FB　　10 | |
| 状态数据 | DB　　30 | DB　　30 | |

**2. 编写程序**

程序的调用结构清晰地显示了主程序中包含哪些功能或者功能块，便于阅读程序。主程序的调用结构如图 13-21 所示。主程序如图 13-22 所示。

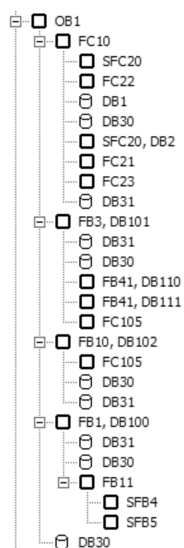
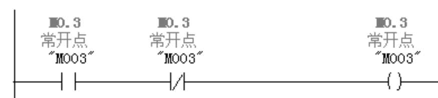
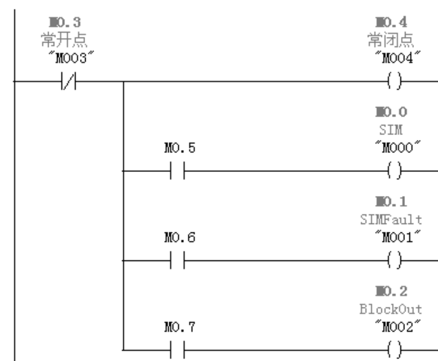



图 13-21　主程序的调用结构

⊟ **程序段 1**：常开点



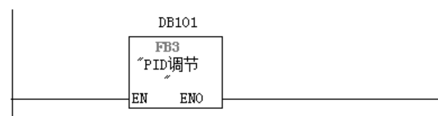

⊟ **程序段 2**：常闭点



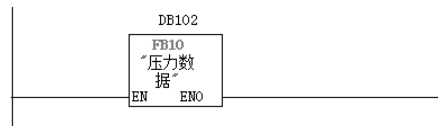

⊟ **程序段 3**：标题：




⊟ **程序段 4**：标题：




⊟ **程序段 5**：标题：




⊟ **程序段 6**：标题：




⊟ **程序段 7**：干粉分配器故障

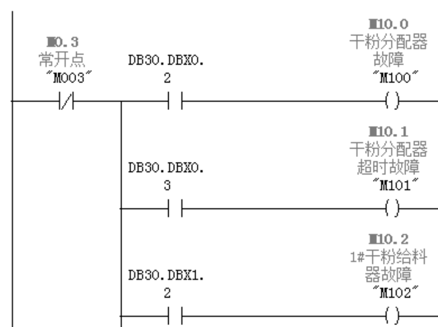



图 13-22　主程序（OB1）

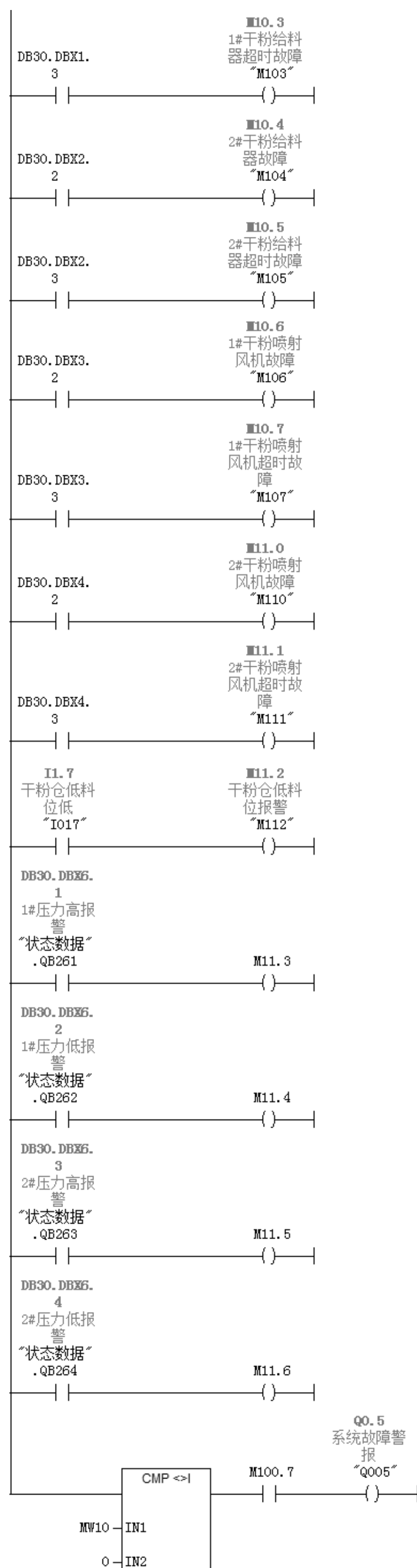



图 13-22　主程序（OB1）（续）

CPU315 -2P 组态为从站模式，在硬件组态时，设置如图 13-23 所示。选中“组态”选项卡，可以看到 CPU315 -2P，接收 DCS 发送来的数据为 IB20 开始的 16 个字节和 IB52 开始的 16 个字节，CPU315 -2P 发送到 DCS 的数据为 QB20 开始的 16 个字节和 QB52 开始的 16 个字节，如图 13-24 所示。DCS 的读功能为 FC10，如图 13-25 所示。




图 13-23　设置从站工作模式




图 13-24　数据存储区组态

功能 FC22 是将 32 位浮点数转化为 16 位整数，如图 13-26 所示。

功能 FC21 是将 16 位整数转化为 32 位浮点数，如图 13-27 所示。

功能 FC23 是将 16 位整数转化为 Time 类型，如图 13-28 所示。

功能块 FB3 如图 13-29 所示。这个功能块的作用是把 HCL 和 $SO_2$ 的浓度进行 PID 运算，再把运算结果传回到 DCS。

功能块 FB10 如图 13-30 所示，这个功能块主要用于测量压力。

⊟ 程序段 1：DCS读取数据

将需要传送给DCS的数据送至专用区域DB1中

M0.3
常开点
"M003"

SFC20
Copy Variables
"BLKMOV"
EN ENO
P#DB30.DBX0.0 BYTE 42 – SRCBLK RET_VAL – LW10
DSTBLK – P#DB1.DBX0.0 BYTE 42

FC22
"Real_TO_Int SUB"
EN ENO
DB30.DBD42 1#干粉给料器频率反馈值 "状态数据".QD62 – Value_Real Value_Int – DB1.DBW8 1#干粉给料器频率反馈 "DCS读数据".QW28
1 – Decimal_Digits

FC22
"Real_TO_Int SUB"
EN ENO
DB30.DBD46 2#干粉给料器频率反馈值 "状态数据".QD66 – Value_Real Value_Int – DB1.DBW10 2#干粉给料器频率反馈 "DCS读数据".QW30
1 – Decimal_Digits

FC22
"Real_TO_Int SUB"
EN ENO
DB30.DBD50 1#干粉压力反馈值 "状态数据".QD70 – Value_Real Value_Int – DB1.DBW12 1#干粉压力反馈值 "DCS读数据".QW32
1 – Decimal_Digits

FC22
"Real_TO_Int SUB"
EN ENO
DB30.DBD54 2#干粉压力反馈值 "状态数据".QD74 – Value_Real Value_Int – DB1.DBW14 2#干粉压力反馈值 "DCS读数据".QW34
1 – Decimal_Digits

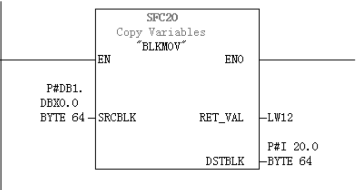
FC22
"Real_TO_Int SUB"
EN ENO
DB30.DBD58 二氧化硫调节反馈值 "状态数据".QD78 – Value_Real Value_Int – DB1.DBW16 二氧化硫调节反馈值 "DCS读数据".QW36
1 – Decimal_Digits

FC22
"Real_TO_Int SUB"
EN ENO
DB30.DBD62 氯化氢调节反馈值 "状态数据".QD82 – Value_Real Value_Int – DB1.DBW18 氯化氢调节反馈值 "DCS读数据".QW38
1 – Decimal_Digits


图 13–25　功能 FC10

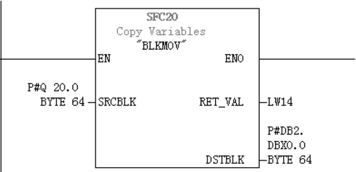
⊟ **程序段 2**：标题：

DCS读取数据（将系统设备状态打包给通讯数据区）




⊟ **程序段 3**：标题：

DCS写入数据（将DCS写入的数据放置在专用区域）




⊟ **程序段 4**：标题：

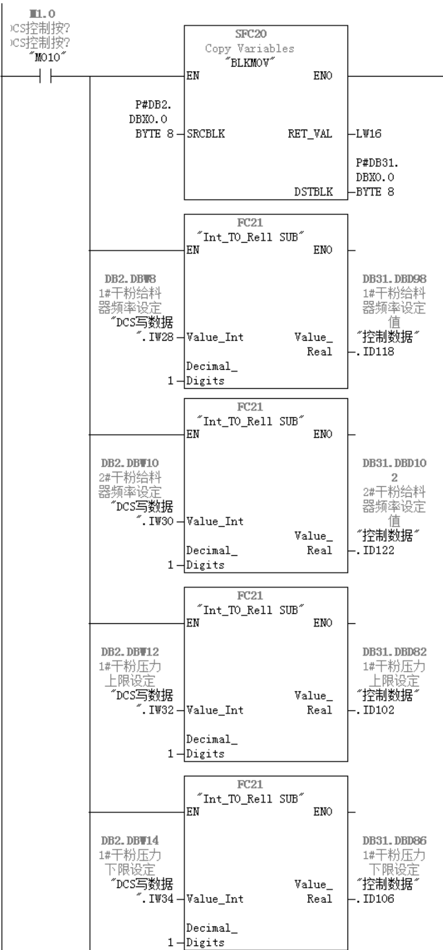



图 13-25　功能 FC10（续）

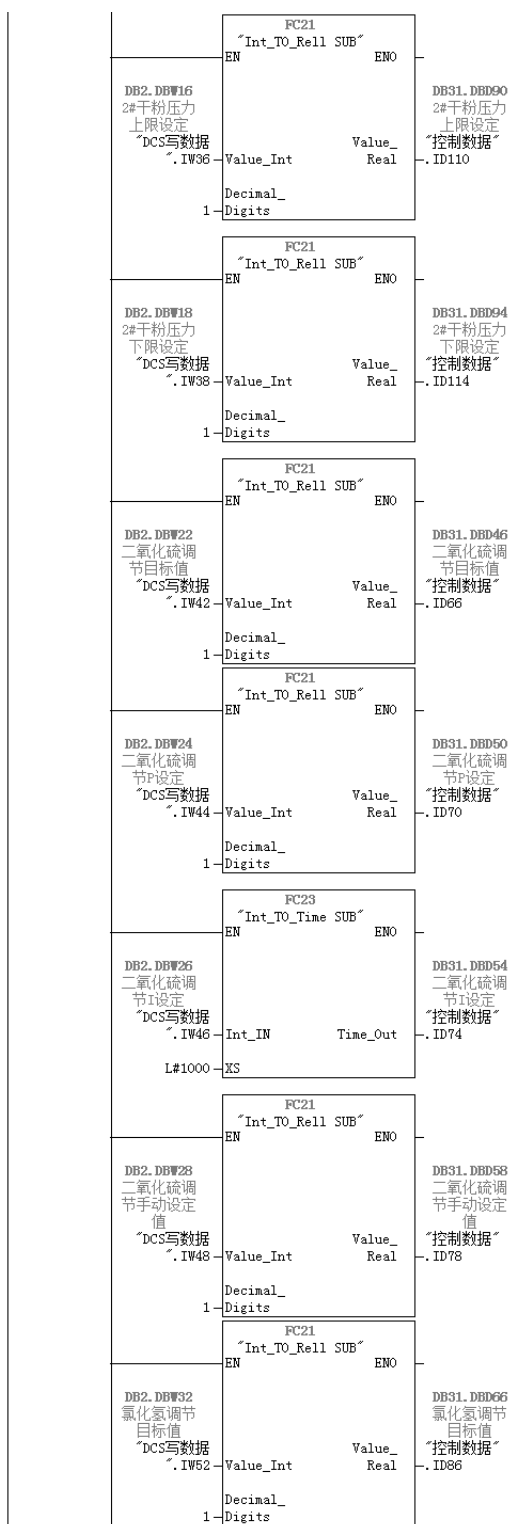
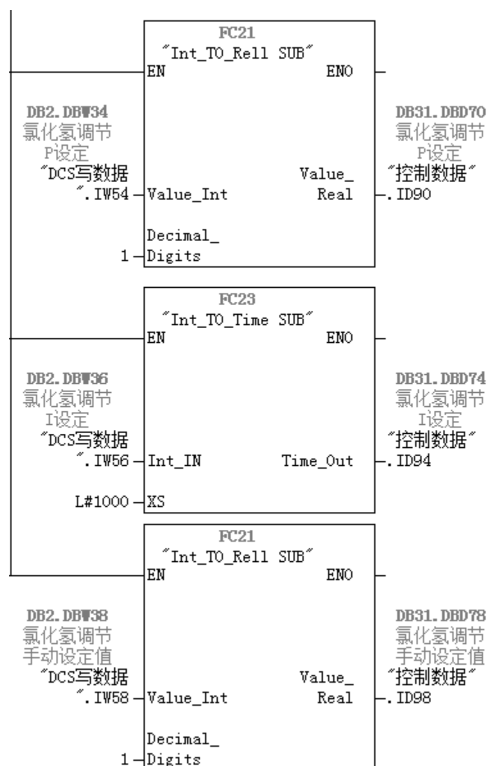
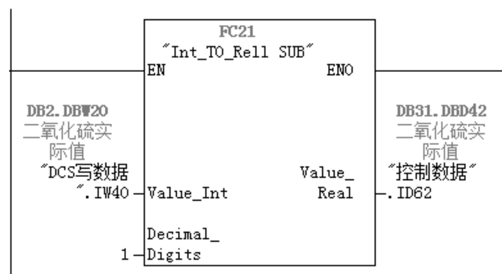


图 13-25　功能 FC10（续）




⊟ **程序段 5**：标题：




⊟ **程序段 6**：标题：

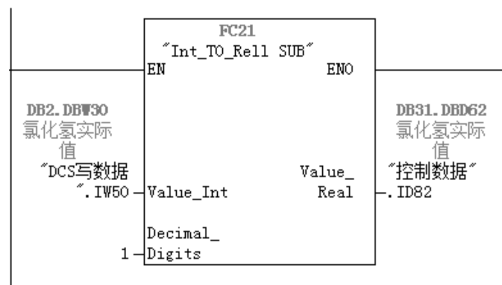
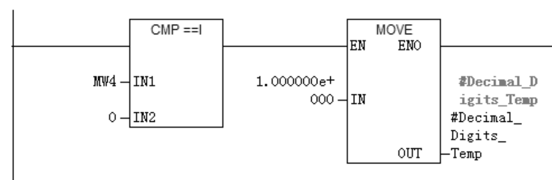
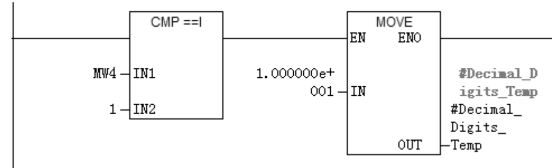



图 13-25　功能 FC10（续）

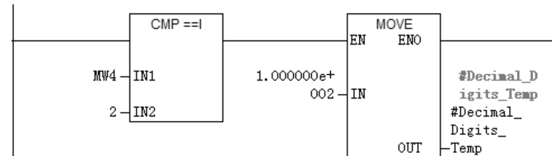
⊟ 程序段 1：标题：



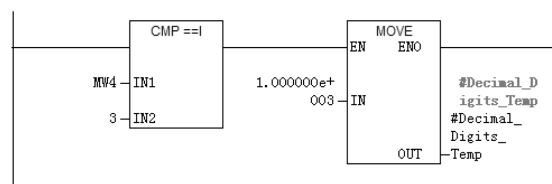

⊟ 程序段 2：标题：



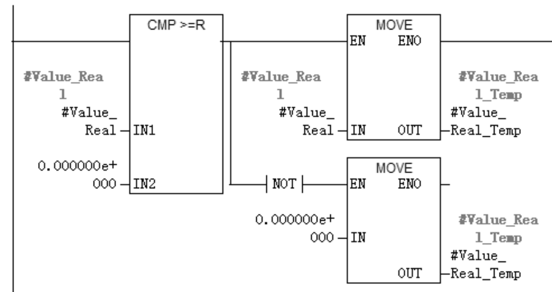

⊟ 程序段 3：标题：




⊟ 程序段 4：标题：




⊟ 程序段 5：标题：




⊟ 程序段 6：标题：

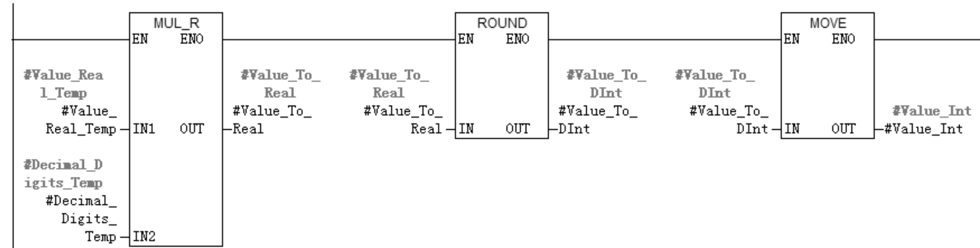



图 13-26　功能 FC22

⊟ 程序段 1：标题：

CMP ==I
#Decimal_D
igits
#Decimal_
Digits — IN1
0 — IN2
MOVE
EN ENO
1.000000e+
000 — IN
OUT — #Decimal_
Digits_
Temp
#Decimal_D
igits_Temp

⊟ 程序段 2：标题：

CMP ==I
#Decimal_D
igits
#Decimal_
Digits — IN1
1 — IN2
MOVE
EN ENO
1.000000e+
001 — IN
OUT — #Decimal_
Digits_
Temp
#Decimal_D
igits_Temp

⊟ 程序段 3：标题：

CMP ==I
#Decimal_D
igits
#Decimal_
Digits — IN1
2 — IN2
MOVE
EN ENO
1.000000e+
002 — IN
OUT — #Decimal_
Digits_
Temp
#Decimal_D
igits_Temp

⊟ 程序段 4：标题：

CMP ==I
#Decimal_D
igits
#Decimal_
Digits — IN1
3 — IN2
MOVE
EN ENO
1.000000e+
003 — IN
OUT — #Decimal_
Digits_
Temp
#Decimal_D
igits_Temp

⊟ 程序段 5：标题：

I_DI
EN ENO
#Value_Int
#Value_Int — IN
OUT — DI
#Value_To_
DI
#Value_To_
DI_R
EN ENO
#Value_To_
DI
#Value_To_
DI — IN
OUT — Real
#Value_To_
Real
#Value_To_
DIV_R
EN ENO
#Value_To_
Real
#Value_To_
Real — IN1
#Decimal_D
igits_Temp
#Decimal_
Digits_
Temp — IN2
OUT — #Value_
Real
#Value_Rea
l

图 13-27　功能 FC21

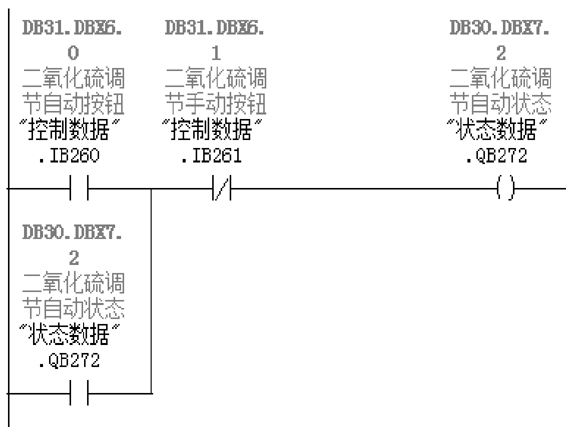
⊟ 程序段 1：标题：

CMP >=I
#Int_IN
#Int_IN — IN1
0 — IN2
CMP <=I
#Int_IN
#Int_IN — IN1
3600 — IN2
I_DI
EN ENO
#Int_IN
#Int_IN — IN
OUT — #Int_DInt_
ms
#Int_DInt_
ms

⊟ 程序段 2：标题：

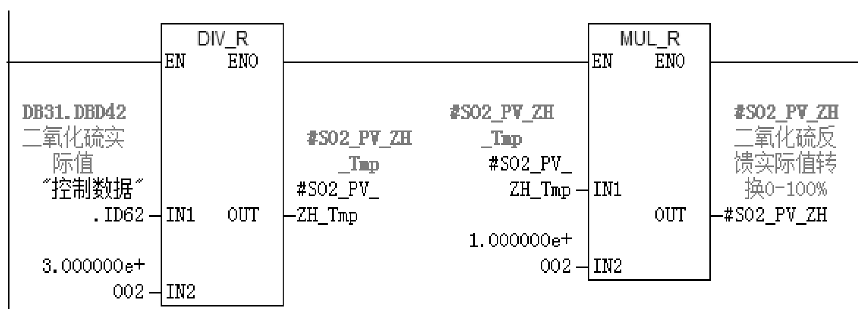
MUL_DI
EN ENO
#Int_DInt_
ms
#Int_DInt_
ms — IN1
#XS
#XS — IN2
OUT — #Int_DInt_
S
#Int_DInt_
S
MOVE
EN ENO
#Int_DInt_
S
#Int_DInt_
S — IN
OUT — #Time_Out
#Time_Out

图 13-28　功能 FC23




⊟ **程序段 2：标题：**

二氧化硫实际值转换




⊟ **程序段 3：标题：**

二氧化硫调节设定值转换

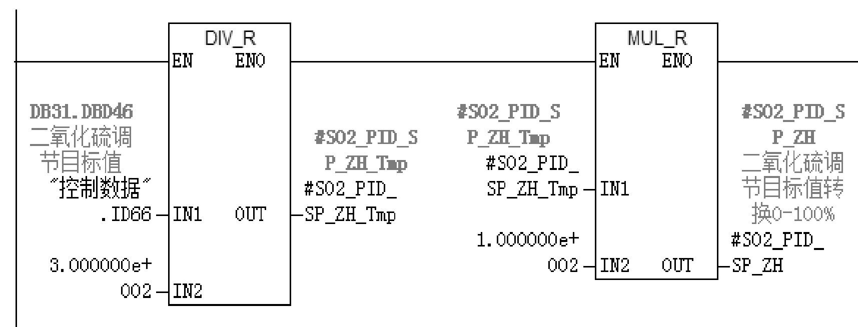



图 13-29　功能 FB3

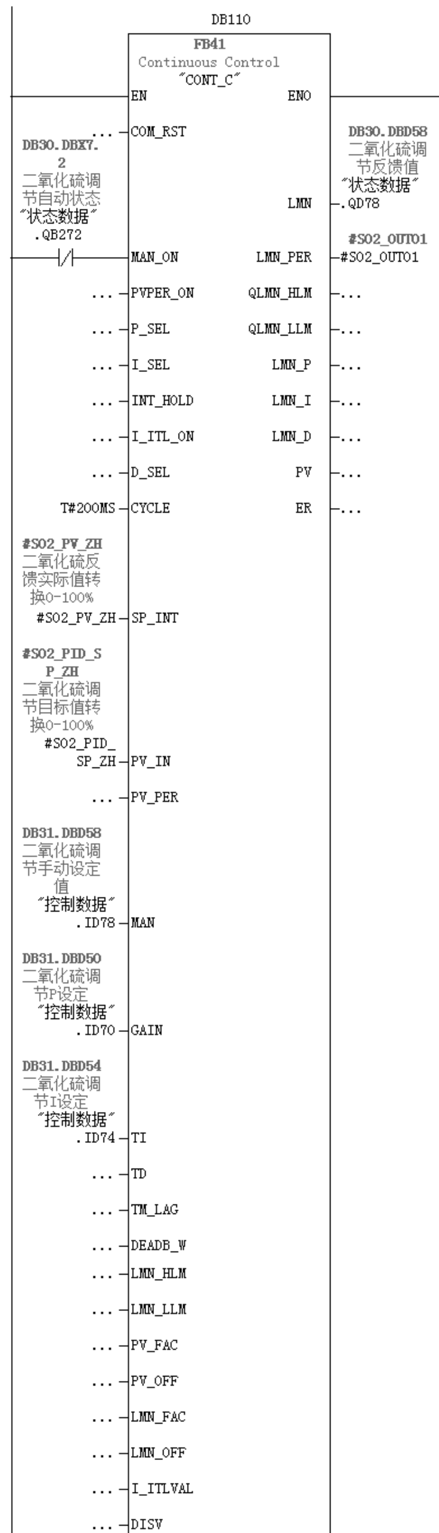
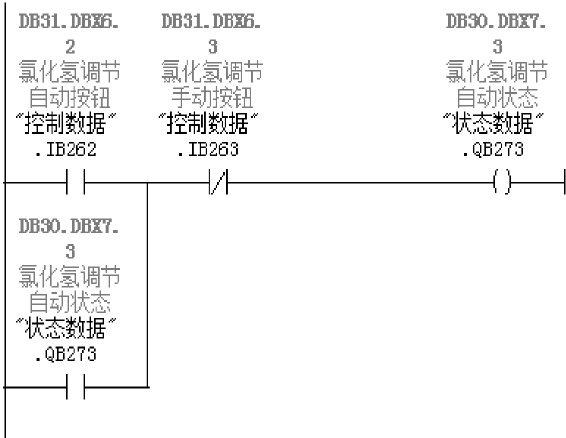


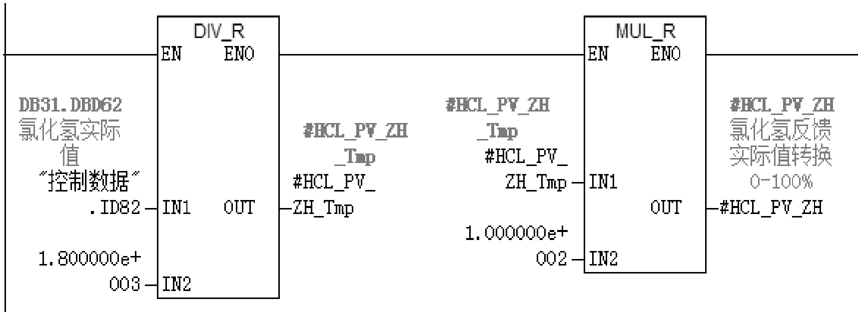

图 13-29　功能 FB3（续）

⊟ **程序段 5：**氯化氢调节自动状态




⊟ **程序段 6：**标题：

氯化氢实际值转换




⊟ **程序段 7：**标题：

氯化氢调节设定值转换，输入氯化氢量程

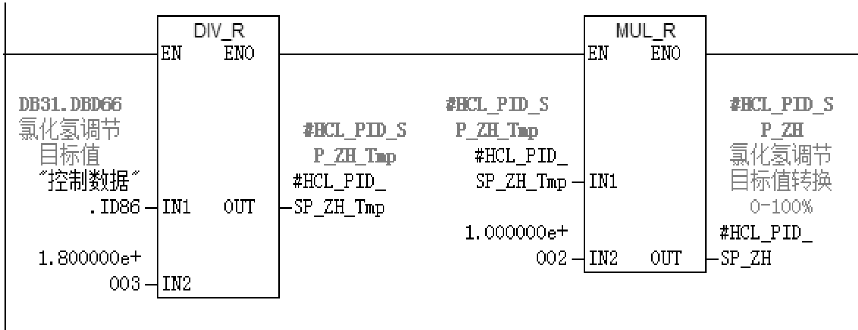



图 13-29　功能 FB3（续）

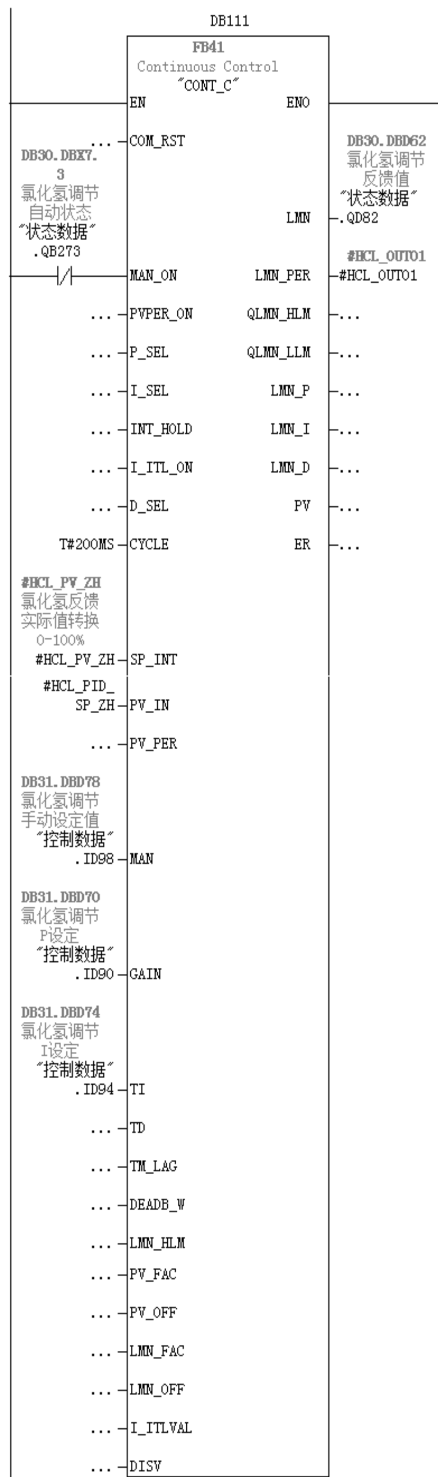
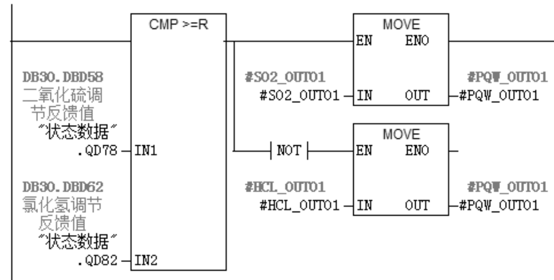
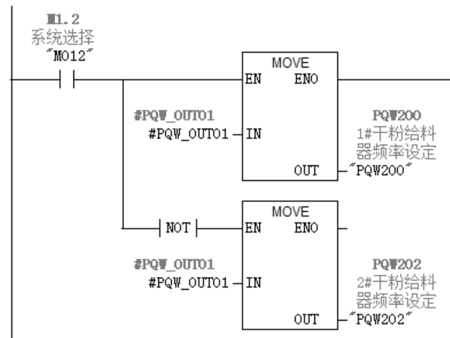



图 13-29　功能 FB3（续）

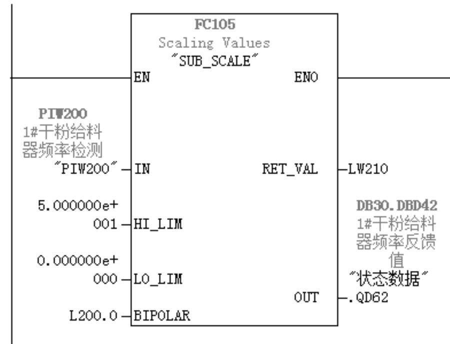
⊟ 程序段 9：标题：




⊟ 程序段 10：标题：




⊟ 程序段 11：标题：




⊟ 程序段 12：标题：

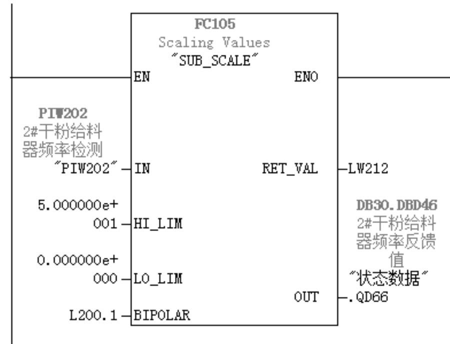
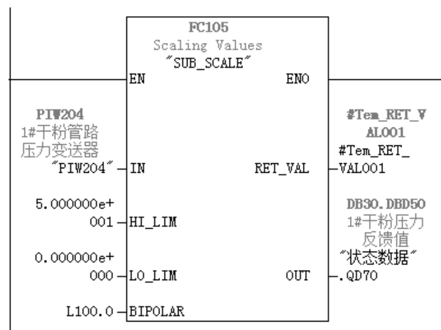
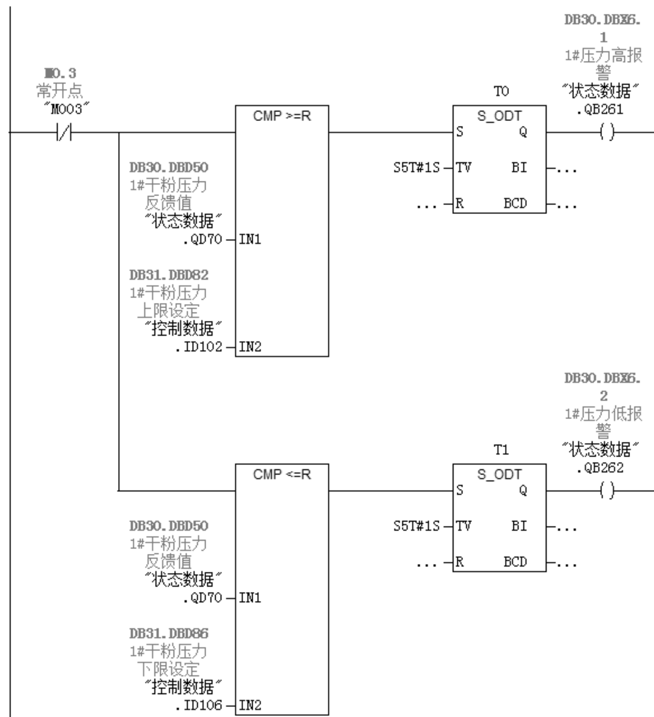



图 13-29　功能 FB3（续）

⊟ 程序段 1：标题：




⊟ 程序段 2：1#压力报警




⊟ 程序段 3：标题：

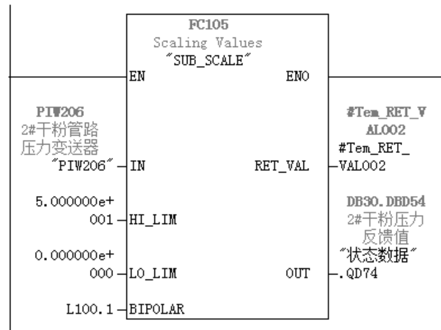



图 13-30　功能 FB10

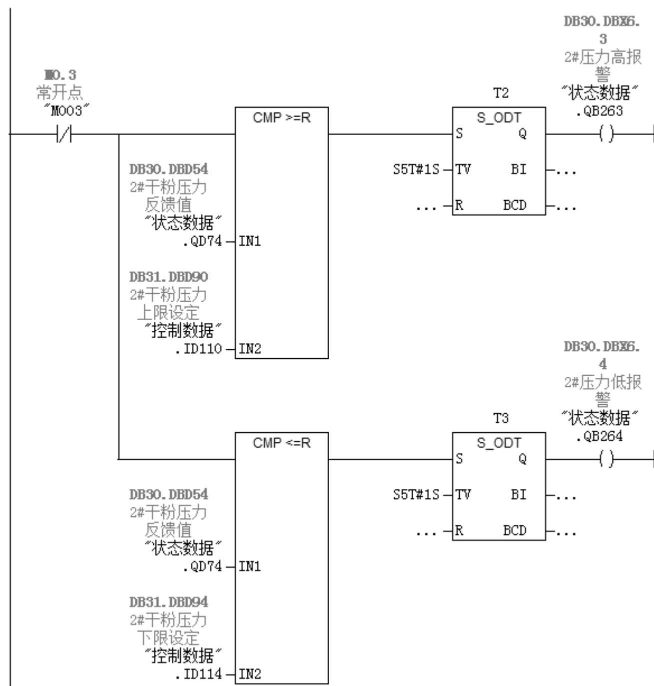
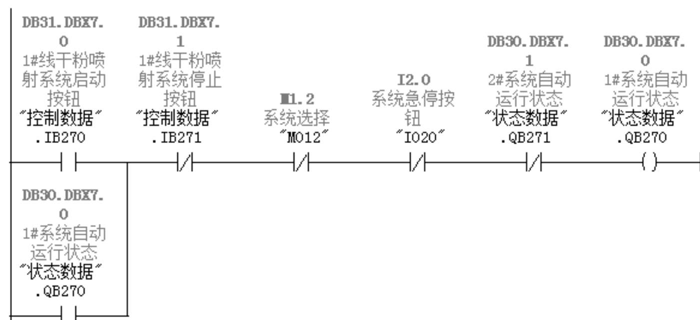



图 13-30　功能 FB10（续）

功能块 FB1 如图 13-31 所示，此功能块中用到多重背景。



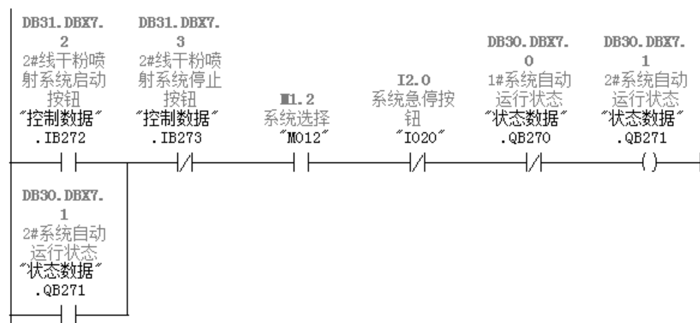
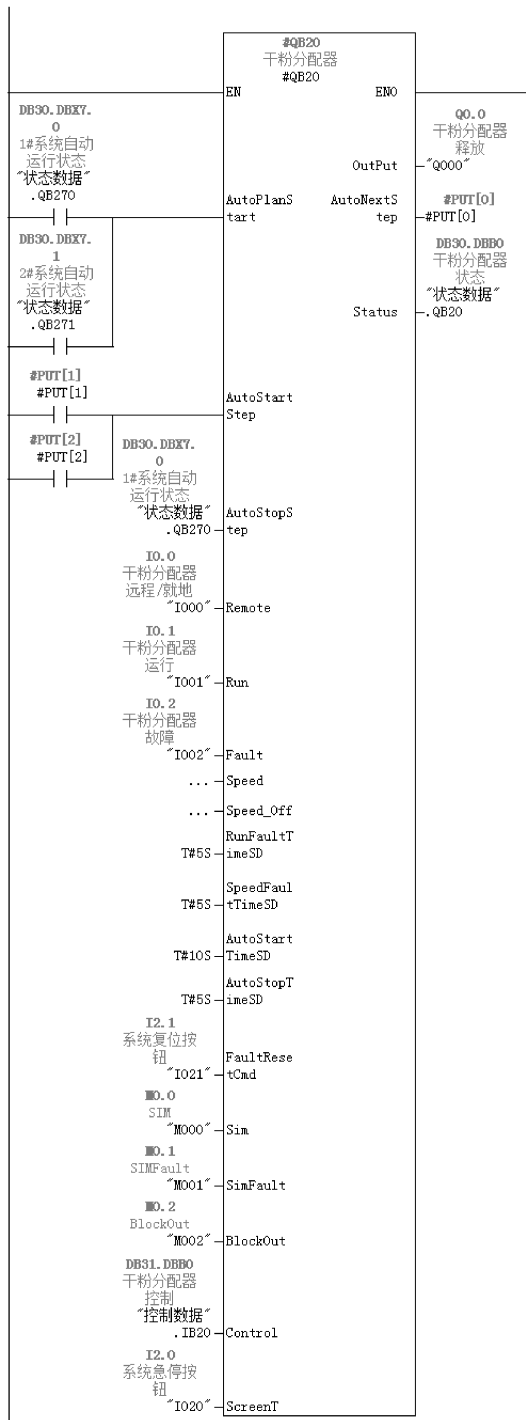




图 13-31　功能 FB1

⊟ **程序段 3**：标题：




⊟ **程序段 4**：标题：

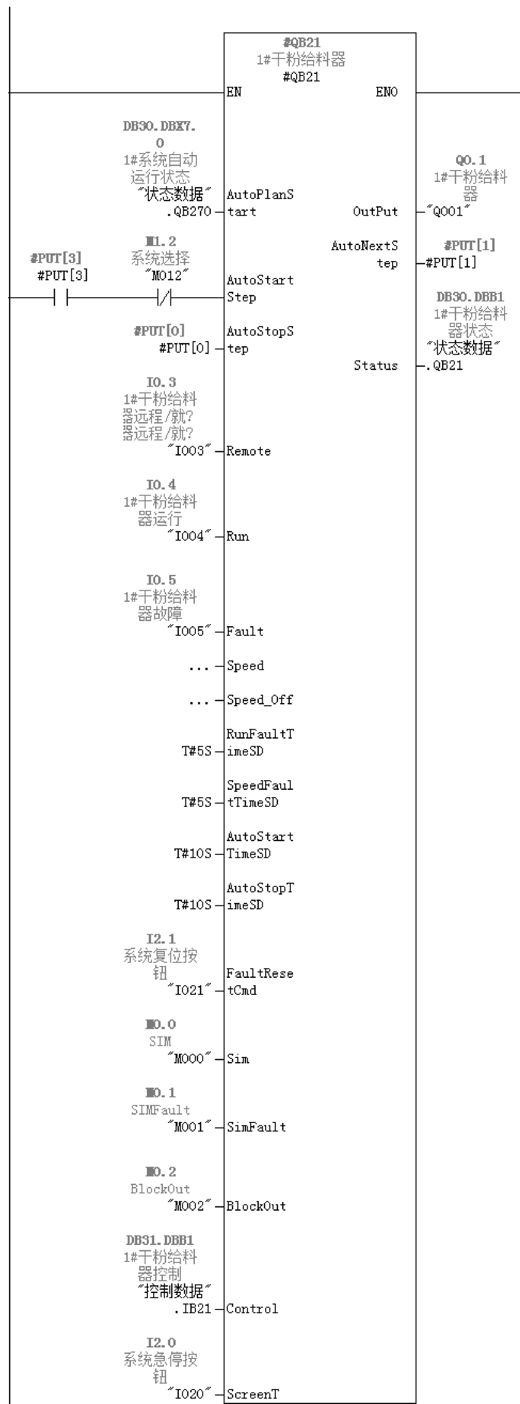
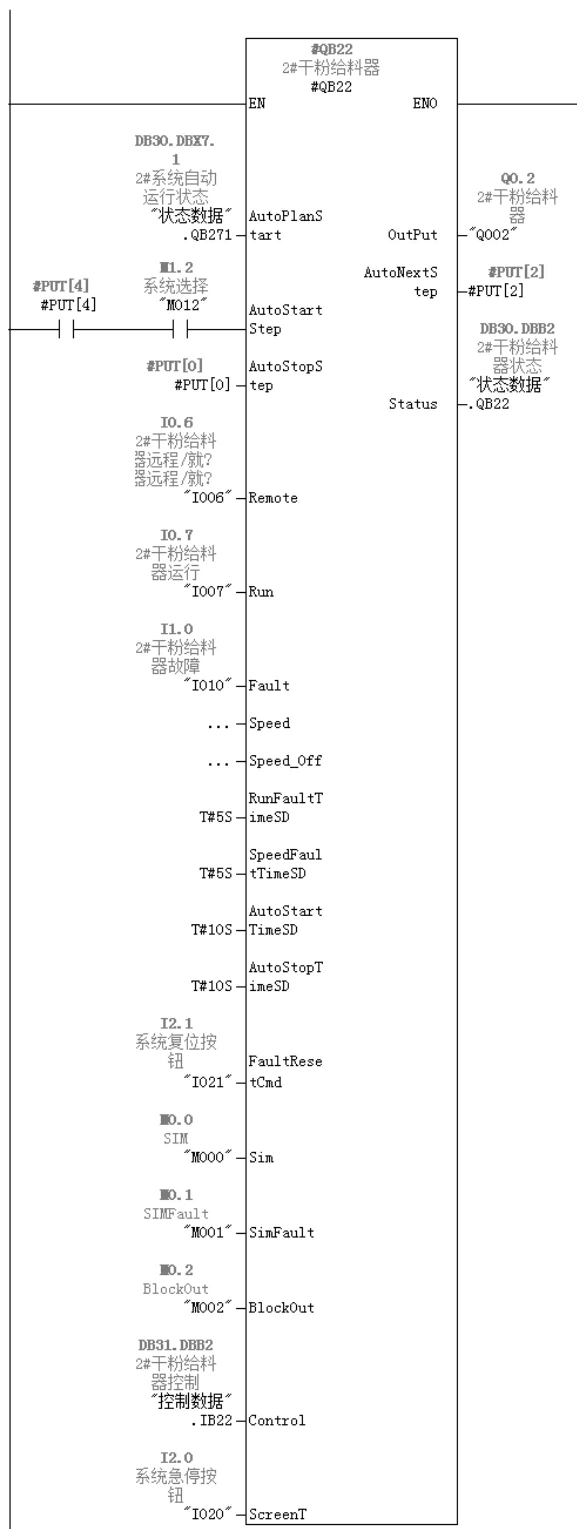



图 13-31　功能 FB1（续）



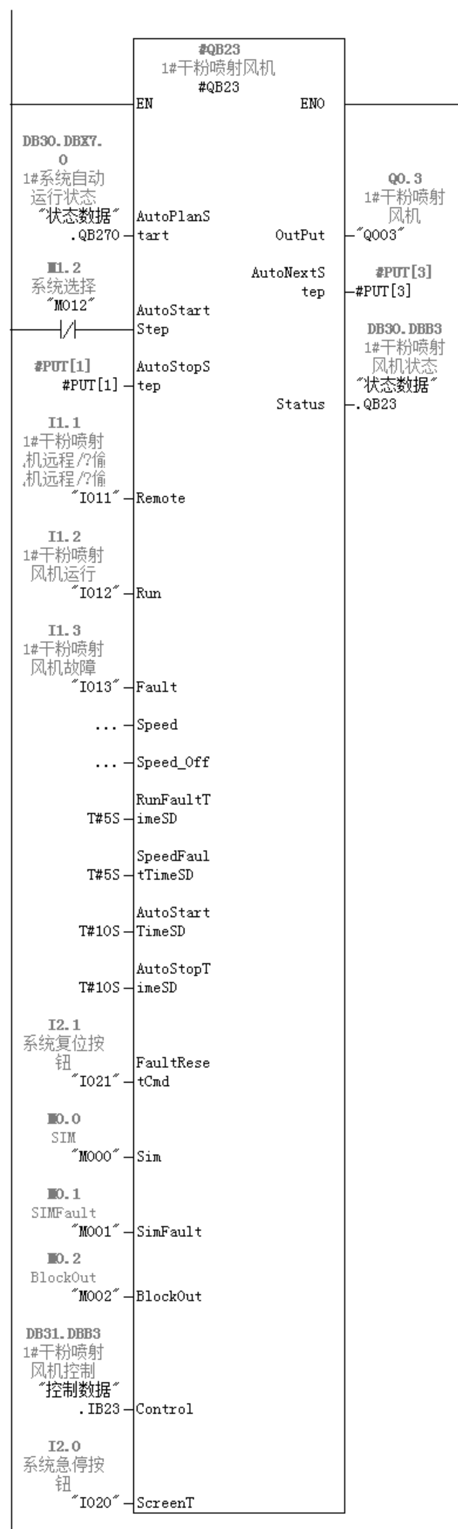
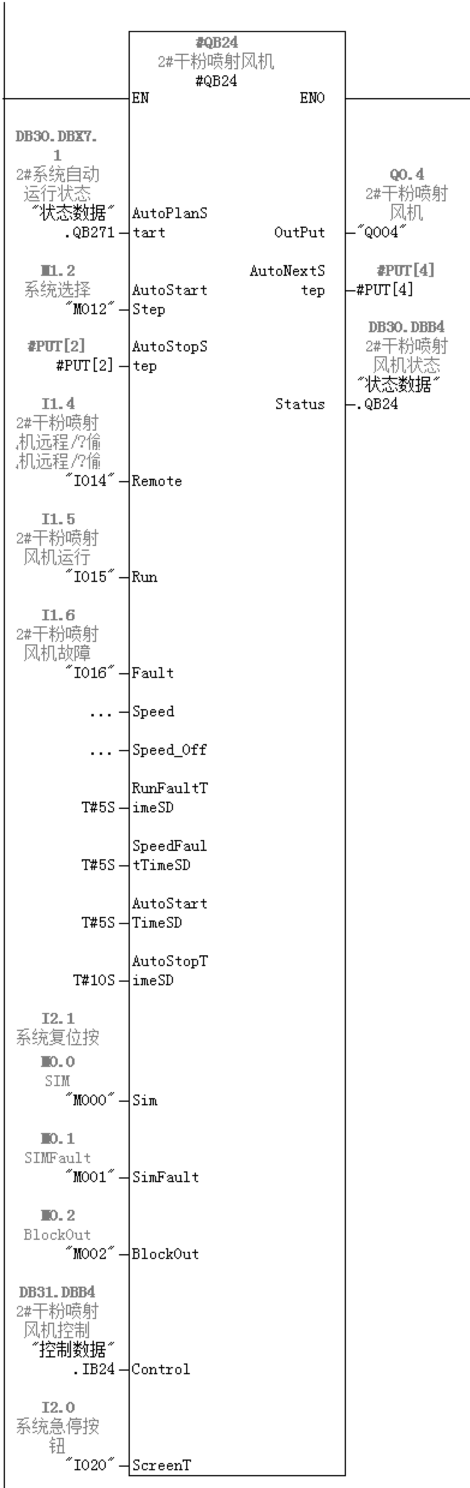




图 13-31　功能 FB1（续）

⊟ **程序段 7**：标题：




⊟ **程序段 8**：标题：

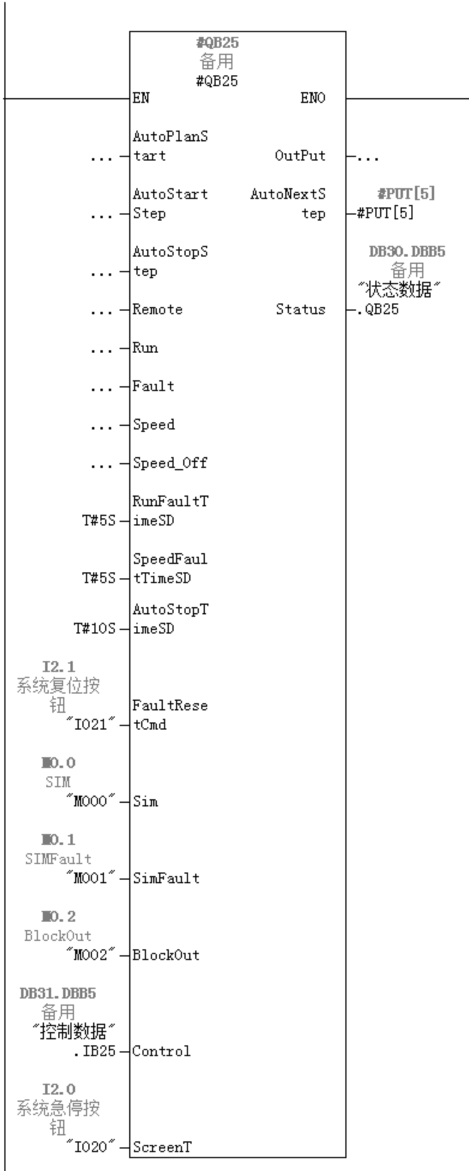



图 13-31　功能 FB1（续）

### 13.2.3 完成触摸屏 HMI 项目

本项目配西门子的 12 吋 MP370 触摸屏，其主画面如图 13-32 所示，主画面有登录密码和用户名称，此密码和用户名称均可修改。

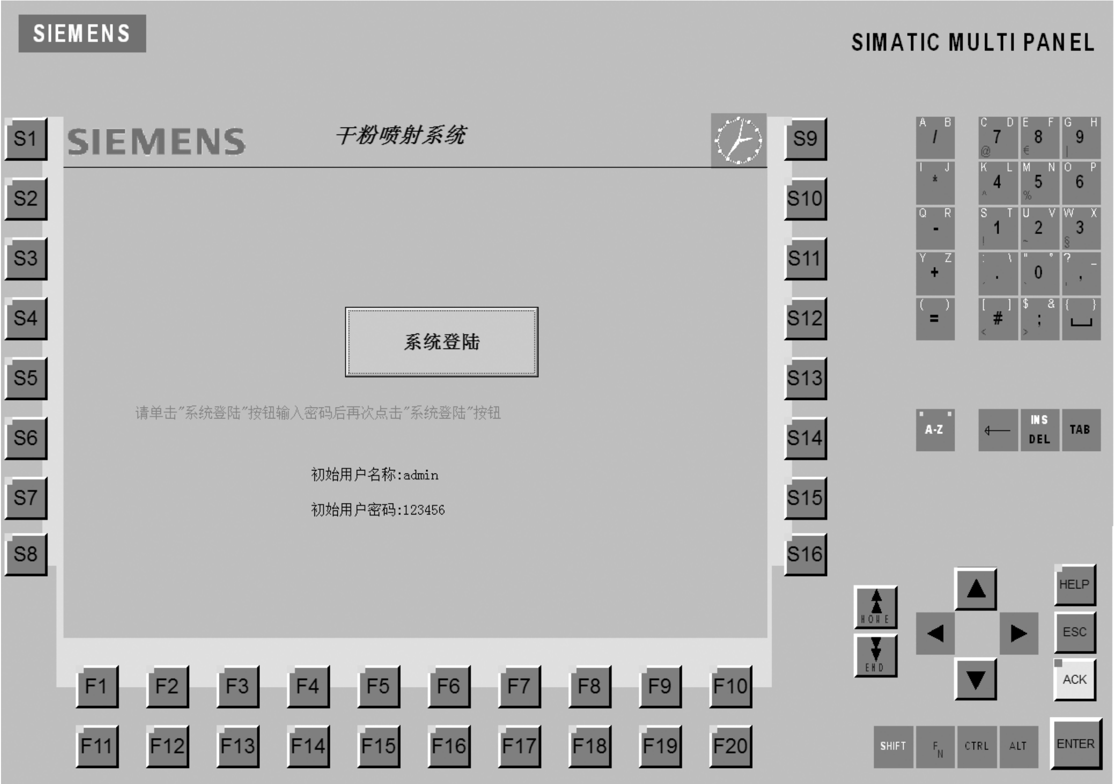



图 13-32 主画面

系统画面如图 13-33 所示，系统画面最为关键，有 $SO_2$ 的浓度 SP（设定值）、PID 调节中的 P（比例）、I（积分）参数和频率等，以上参数都需要设置。HCL 的参数设定与 $SO_2$ 类似。1 号系统的压力上限和下限以及 2 号系统的上限和下限也需要设定。可显示 1 号系统的压力和 2 号系统的压力实时数值。

单机操作画面如图 13-34 所示，在这个界面可以分别对两套系统进行“手动”和“自动”操作。

离散报警变量如图 13-35 所示，有干粉分配器故障、干粉分配器超时故障以及 1#压力高报警等共 15 个报警变量。当报警变量条件满足时，触发如图 13-36 所示的报警画面，在这个界面显示是何种报警，操作和维修人员可根据此报警进行处理或者维修。例如，当报警变量 M10.0 =1 时，满足触发“干粉分配器故障”的条件，此时弹出报警画面，画面中显示“干粉分配器故障”。

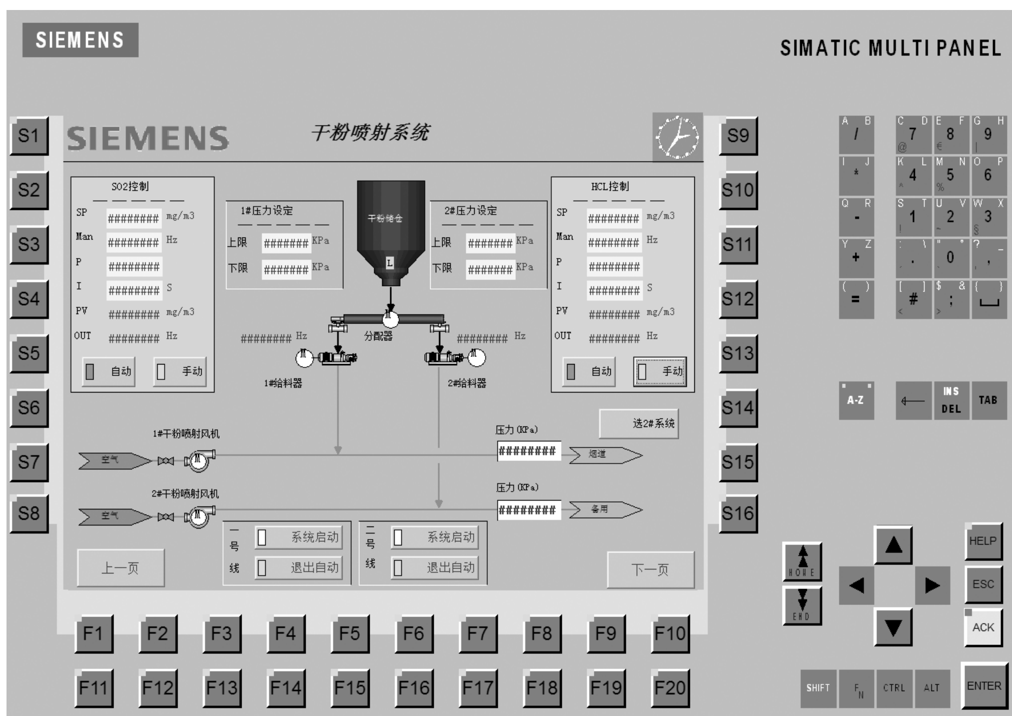



图 13-33　系统画面

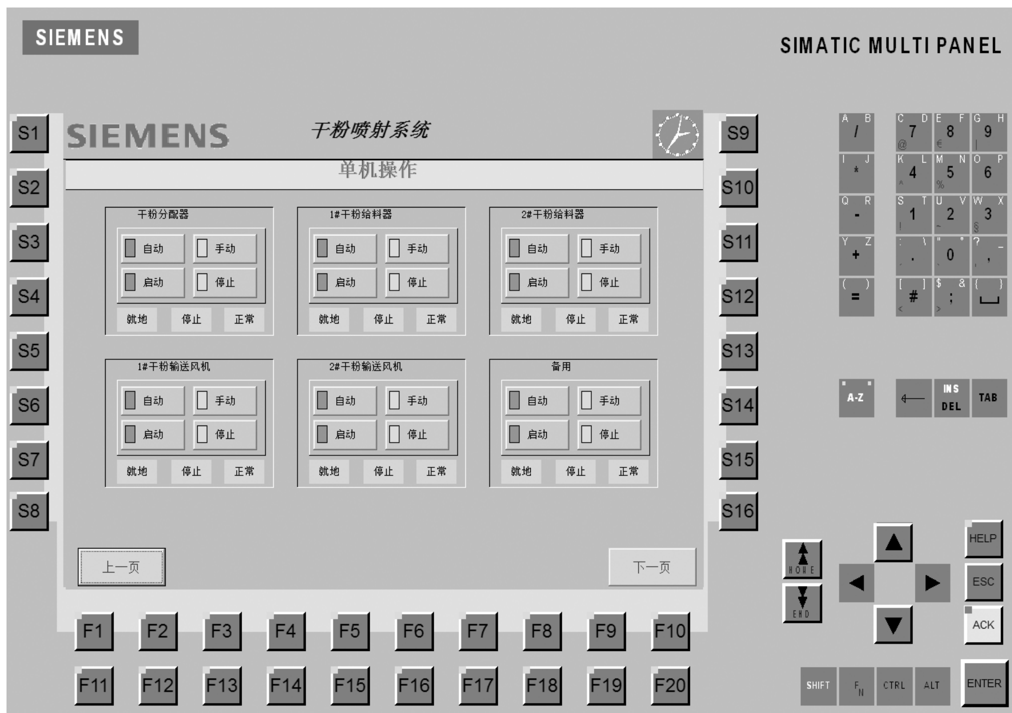



图 13-34　单机操作画面

离散量报警

| 文本 | 编号 | 类别 | 触发变量 | 触发器位 | 触发器地址 |
|---|---|---|---|---|---|
| 干粉分配器故障 | 1 | 错误 | MW10 | 8 | M 10.0 |
| 干粉分配器超时故障 | 2 | 错误 | MW10 | 9 | M 10.1 |
| 1#干粉给料器故障 | 3 | 错误 | MW10 | 10 | M 10.2 |
| 1#干粉给料器超时故障 | 4 | 错误 | MW10 | 11 | M 10.3 |
| 2#干粉给料器故障 | 5 | 错误 | MW10 | 12 | M 10.4 |
| 2#干粉给料器超时故障 | 6 | 错误 | MW10 | 13 | M 10.5 |
| 1#干粉喷射风机故障 | 7 | 错误 | MW10 | 14 | M 10.6 |
| 1#干粉喷射风机超时故障 | 8 | 错误 | MW10 | 15 | M 10.7 |
| 2#干粉喷射风机故障 | 9 | 错误 | MW10 | 0 | M 11.0 |
| 2#干粉喷射风机超时故障 | 10 | 错误 | MW10 | 1 | M 11.1 |
| 干粉仓低料位报警 | 11 | 错误 | MW10 | 2 | M 11.2 |
| 1#压力高报警 | 12 | 错误 | QW6 | 9 | DB 30 DBX 6.1 |
| 1#压力低报警 | 13 | 错误 | QW6 | 10 | DB 30 DBX 6.2 |
| 2#压力高报警 | 14 | 错误 | QW6 | 11 | DB 30 DBX 6.3 |
| 2#压力低报警 | 15 | 错误 | QW6 | 12 | DB 30 DBX 6.4 |

图 13-35　离散报警变量

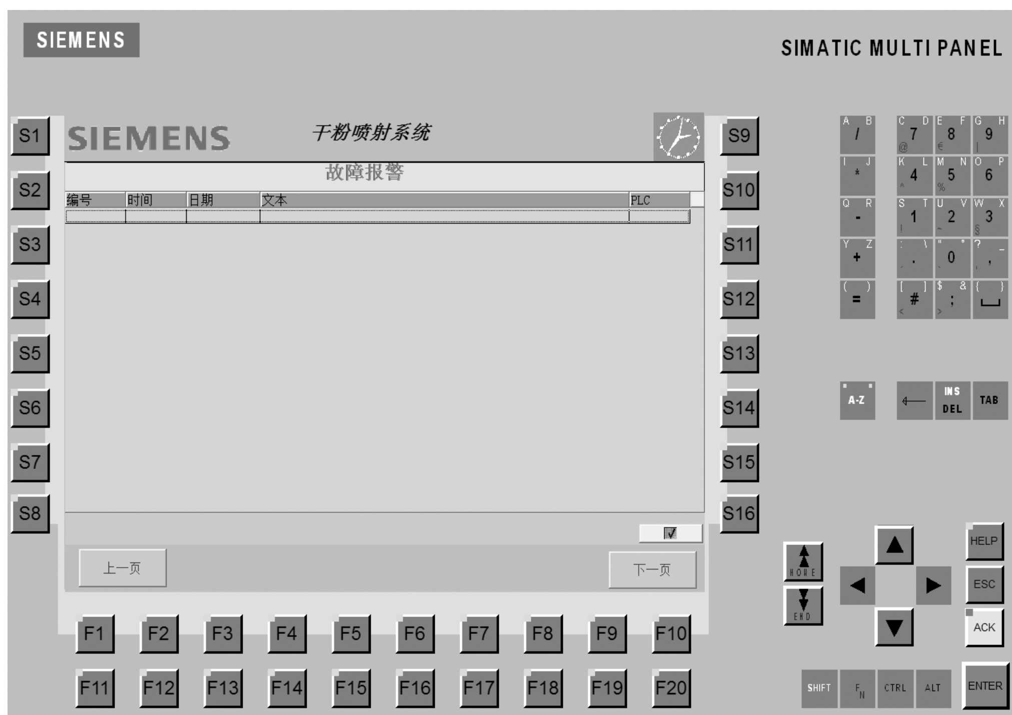



图 13-36　报警画面

# 参 考 文 献


[1] 向晓汉．西门子 S7－300 PLC 完全精通教程[M]．北京：化学工业出版社，2016.
[2] 向晓汉，陆彬．电气控制与 PLC 技术[M]．北京：人民邮电出版社，2012.
[3] 刘楷．深入浅出 西门子 S7－300 PLC[M]．北京：北京航空航天大学出版社，2004.
[4] 柴瑞娟．西门子 PLC 编程技术及工程应用[M]．北京：机械工业出版社，2007.
[5] 张运刚．从入门到精通－西门子 S7－300/400PLC 技术与应用[M]．北京：人民邮电出版社，2007.
[6] 秦益霖．西门子 S7－300PLC 应用技术[M]．北京：电子工业出版社，2007.
[7] 廖常初．S7－300/400PLC 应用技术[M]．北京：机械工业出版社，2013.
[8] 向晓汉．西门子 PLC 工业网络完全精通教程[M]．北京：化学工业出版社，2013.
[9] 崔坚．西门子工业网络通信指南[M]．北京：机械工业出版社，2009.
[10] 崔维群，孙启发．S7－300/400 可编程控制器原理与应用[M]．北京：北京航空航天大学出版社，2009.
[11] 西门子（中国）有限公司．S7－200 可编程控制器系统手册[Z]，2008.
[12] 西门子（中国）有限公司．MICROMASTER 440 标准变频器使用大全[Z]，2011.

在线互动交流平台

官方微博：http://weibo.com/cmpjsj
机工工控书友群：495046628

# 西门子 PLC、变频器、触摸屏工程应用及故障诊断



## 本书特色

- 用实例引导读者学习。本书的大部分章节用精选的例子讲解。例如，用例子说明现场总线通信的实现的全过程。
- 重点的例子都包含软硬件的配置方案图、接线图和程序，而且为确保程序的正确性，程序已经在 PLC 上运行通过。
- 对于比较复杂的例子，配有录像资料。如工业以太网通信的硬件组态较复杂，就配有组态过程的录像资料，便于读者学习。
- 实用性强，本书的实例易于进行工程移植。

地址：北京市百万庄大街22号
邮政编码：100037
电话服务
社服务中心：010-88361066
销售一部：010-68326294
销售二部：010-88379649
读者购书热线：010-88379203
网络服务
教材网：http://www.cmpedu.com
机工官网：http://www.cmpbook.com
机工官博：http://weibo.com/cmp1952
封面无防伪标均为盗版


机械工业出版社微信服务号


计算机分社微信服务号


机工社电气信息图书公众号

上架指导 工业技术/PLC
ISBN 978-7-111-58358-5
策划编辑◎ 时 静 / 封面设计◎







定价：129.00 元